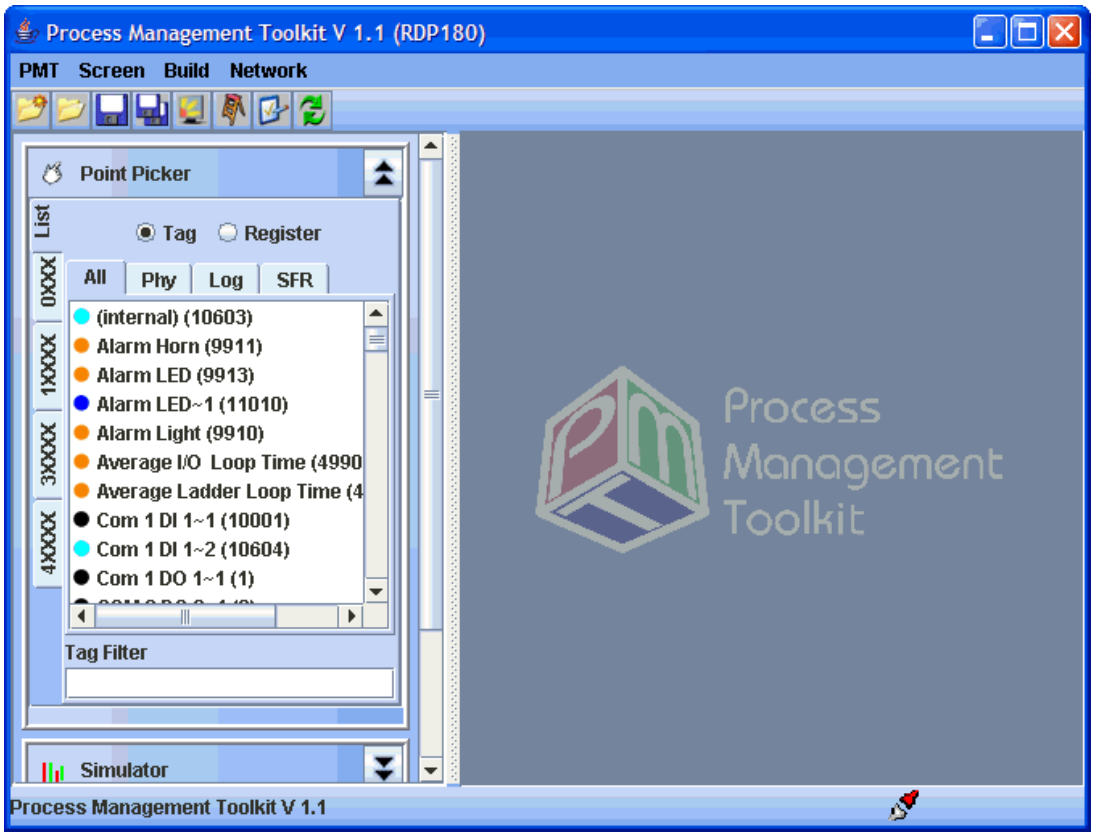
Data Flow Systems



PROCESS MANAGEMENT TOOLKIT

USER MANUAL

PROCESS MANAGEMENT TOOLKIT
USER MANUAL

605 N. John Rodes Blvd., Melbourne, FL 32934
Phone 321-259-5009 • Fax 321-259-4006

NOTICE

Data Flow Systems, Inc. assumes no responsibility for any errors that may appear in this document, nor does it make any commitment to update the information contained herein. However, questions regarding the information contained in this document are welcomed.

Data Flow Systems also reserves the right to make changes to the specifications of the Process Management Toolkit and to the information contained in this document at any time without notice.

DFS-00509-011

This document last revised: August 23, 2011

TABLE OF CONTENTS

PURPOSE OF THIS MANUAL

This manual is a reference guide for the Process Management Toolkit (PMT). The PMT is a software suite used to configure and map I/O, program devices using ladder logic, and build custom user interfaces (screens). The PMT is designed for use with both Data Flow Systems' PLC033 and Open Control Solutions' RDP033 and RDP180.

DOCUMENT CONVENTIONS

The following conventions are used throughout this manual:

- Unless otherwise noted, the term PLC is used when referring to either the PLC033 or the RDP.
- Bulleted lists provide information, not procedural steps.
- Numbered lists provide sequential steps or hierarchal information.
- *Italic* type is used for emphasis

ABBREVIATIONS USED IN THIS MANUAL

ACM – Analog Control Module

AMM – Analog Monitor Module

I/O – Input/Output

BEM – Bus Extender Module

DCM – Digital Control Module

DMM – Digital Monitor Module

FIM – Fiber Interface Module

LSB – Least Significant Bit

MBP – Modular Backplane

MSB – Most Significant Bit

PLC – Programmable Logic Controller

PMT – Process Management Toolkit

RDP – Rail Data Processor

RIM – Radio Interface Module

RIO – Rail I/O (RIO128 with 128 I/O; RIO032 with 32 I/O)

RTU – Remote Terminal Unit

Notes

Chapter 1: PRODUCT OVERVIEW

DESCRIPTION

The Process Management Toolkit (PMT) is a software suite used to configure and program the PLC033, the RDP033, and the RDP180.

It features the following applications:

- **I/O Builder**: Configure network settings; set up communication parameters for the PLC's serial ports. Add and configure the devices - and their associated physical I/O - that the PLC will interface with, including DFS modules (local and remote) and Modbus master and slave devices.
- **Mapper**: Map all physical I/O to the correct register range in the PLC's Logical Memory Map. By mapping all points, both DFS-type points and Modbus-type points, to a common map using a common format, you are able to easily create logic for devices that use different communication protocols. If the PLC is to be polled by an HT3 system using a DFS protocol (RIM protocol for radio link; NIM RTU protocol for network interface), you will also use this utility to map the Logical Memory Map registers into up to 30 DFS modules in the Radio Map [up to 15 modules per station with a two (2) station maximum].
- **Screen Builder**: Create user interfaces to the PLC and the devices connected to it. These screens can show you the status of a device (is the pump running or off? what is the tank's level?). They can also be used to control a device (click a button to turn a pump on or off; enter a value in an input field to force the tank to the specified level).
- **Logic Builder**: Create the logical, ladder-based programs that run on the PLC.
- **Network Tools:** Connect to and disconnect from the PLC over the network; test network and serial communications.
- **Simulator:** Enables you to simulate system activity when disconnected from the PLC. When you are connected to the PLC, this utility behaves as an animator showing the real values coming from the PLC and allowing you to send control messages to the PLC.

I/O MAPPING

The PLC is a Modbus device. Modbus is an industry-standard protocol and as such enables the PLC to communicate with any third-party Modbus devices, including Open Control Solutions' RIO128 and RIO032 rail-mounted I/O devices. The PLC is able to support both DFS and Modbus protocols through a process called mapping.

In mapping, each physical I/O point is assigned to a unique register in the PLC's Logical Memory Map. The Logical Memory Map is comprised of four register ranges:

I/O Type	Register Range
Digital Outputs (Coils)	00001-09999
Digital Inputs (Discrete Inputs)	10001-19999
Analog Inputs (Input Registers)	30001-39999
Analog Outputs (Holding Registers)	40001-49999

The physical I/O of all devices connected to the PLC (local DFS function modules as well as the I/O of any external Modbus-compatible devices) must be mapped. This ensures that there are no duplicate, or conflicting, addresses. Once mapping is complete, these registers are available for use in ladders and custom screens.

When doing this type of mapping, you need to be aware of the registers that have been set aside for special functions (for example, controlling the four programmable LEDs) and the registers reserved for set points. A list of these registers can be found in the sections "Reserved Set Point Registers" and "Special Function Registers," below and on the next page.

A second form of mapping, DFS Radio mapping, is required in order for the PLC to be polled by an HT3 system using a DFS protocol (RIM protocol – radio link; NIM RTU protocol – network interface). In this process, the Logical Memory Map registers are mapped to up to 30 DFS modules in the Radio Map [up to 15 modules per station with a two (2) station maximum]. For example, registers in the 00001-09999 range (digital outputs) would be mapped to one or several Digital Control Modules (DCM003); registers in the 40001-49999 range (analog outputs) would be mapped to one or several Analog Control Modules (ACM002). The mapped I/O can be any combination of physical I/O, logical I/O generated by ladder programs, and special function registers.

Because the DFS RIM and NIM RTU protocols are 12-bit protocols with a full-scale output of 4095, you must specify how you want to shift the bits in the register when working with any data that is greater than 12 bit that will be sent over a DFS radio link or network interface (for example, 15-bit RIO128 data or 16-bit logical data). Details on this process can be found in the section titled "DFS Radio Mapping: Register (Bit) Shifting Options" on page 57.

RESERVED SET POINT REGISTERS

Blocks of registers in the DO and AO ranges are reserved for storing user set point values. Values stored in these registers are automatically written (saved) to the PLC's flash memory every 30 seconds. They are also saved during a controlled shutdown.

- During a controlled shutdown (see next section), user set point values are written to the PLC's flash memory. These stored values are loaded when the process resumes.
- In the event of an abrupt loss of power, the process will use the values written to the PLC's flash memory during the last automated save.

The following registers are reserved for set point values:

- Registers 9800-9899 are reserved in the DO range.
- Registers 49800-49899 are reserved in the AO range.

CONTROLLED SHUTDOWN

The following events initiate a controlled shutdown:

- If voltage in the panel falls to 11.0 volts, the PLC's operating system will initiate the shutdown without user intervention.
- Press the shutdown button on the PLC to manually stop the process.
- Use Special Function Registers – Remote Process Start Command (9920) and Remote Process Stop Command (9921) – to shutdown the process from a remote location. For example, you could map

these registers to a DFS module point in the PLC's DFS Radio Map and configure the same point in HT3. The registers could then be controlled from a custom screen or a default screen.

SPECIAL FUNCTION REGISTERS

The following local and derived I/O points are provided as special function registers. Most of these registers are for external status queries. For example, registers 9930 (Process Running) and 49902 (Max Ladder Loop Time) give you important information on the status of the PLC. Others, such as 9910 (RDP Output 1 [PLC LED DS12]) and 9913 (Alarm LED [PLC LED DS09]) can be used in ladders to control the behavior of the PLC.

NOTES:

- Registers specific to the RDP033/180 include the word "RDP" in their name. For example, RDP Input 7.
- Registers specific to the PLC033 include the word "PLC" in their name and are contained in brackets. For example, [PLC Central Mode].
- The names of registers with functions that are different depending on the type of device being configured (RDP033/180 or PLC033) are shown with the RDP function name followed by brackets containing the PLC033 function name. For example, register 9900 is listed as RDP Input 1 [PLC Config Bit 0].
- Registers common to both the PLC033 and the RDP033/180 are listed normally.
- Registers 9997 and 9998 are used to make sure the RDP and Dataport programs are always running; they are not for use in ladders or at remotes. One program continually updates the register while the other watches it. Should updating stop for 60 seconds after the initial update, the other program will reboot the device.
- Registers 49945 (Cell Signal Strength), 49946 (Cell Connection Count), and 49947 (Cell Error Count) are used to store cell modem connection data for Cellular RTUs.

Table 1-1, Special Function Registers

Address	RDP Function [PLC Function]	Ladder R/W	Remote R/W
9900	RDP Input 1 [PLC Config Bit 0]	R	R
9901	RDP Input 2 [PLC Config Bit 1]	R	R
9902	RDP Input 3 [PLC Config Bit 2]	R	R
9903	RDP Input 4 [PLC Config Bit 3]	R	R
9904	RDP Input 5 [PLC Config Bit 4]	R	R
9905	RDP Input 6 [PLC Config Bit 5]	R	R
9906	RDP Input 7	R	R
9907	RDP Input 8	R	R
9908	RDP Input 9	R	R
9910	RDP Output 1 [PLC LED DS12]	RW	R
9911	RDP Output 2 [PLC LED DS11]	RW	R
9912	Hook LED [PLC LED DS10]	RW	R
9913	Alarm LED [PLC LED DS09]	RW	R
9920	Process Start Command	R	RW

Address	RDP Function [PLC Function]	Ladder R/W	Remote R/W
9921	Process Stop Command	R	RW
9930	Process Running	RW	R
9940	Global Alarm	R	R
9941	Low Memory Alarm	R	R
9942	Low Voltage Alarm	R	R
9943	RDP I/O Fault Alarm	R	R
9950	[PLC Central Mode]	RW	R
9997	RDP Updates	--	--
9998	Dataport Updates	--	--
49900	Power Supply Voltage	R	R
49901	OS Free Memory	R	R
49902	Max Ladder Loop Time	R	R
49903	Avg Ladder Loop Time	R	R
49904	Min Ladder Loop Time	R	R
49908	Hardware Model	R	R
49909	Hardware Revision	R	R
49910	Software Version Year	R	R
49911	Software Version Month	R	R
49912	Software Version Day	R	R
49913	OS Version Year	R	R
49914	OS Version Month	R	R
49915	OS Version Day	R	R
49916	Serial Number High	R	R
49917	Serial Number Low	R	R
49918	Network Address Octet 4	R	R
49919	Ladder Process ID	RW	R
49920	Ladder Version Year	RW	R
49921	Ladder Version Month	RW	R
49922	Ladder Version Day	RW	R
49923	RDP Analog Input 1	R	R
49945	Cell Signal Strength	R	R
49946	Cell Connection Count	R	R
49947	Cell Error Count	R	R
49950	Comm1 Last Comm	RW	R
49951	Comm2 Last Comm	RW	R
49952	Comm3 Last Comm	RW	R
49953	Comm4 Last Comm	RW	R
49954	Comm5 Last Comm	RW	R
49955	Comm6 Last Comm	RW	R
49999	Remote Reset (data must be 0xA5A5)	R	RW

Q POINTS (PLC033 ONLY)

Q points, named as such because they always reside at module address Q in HT3's Configuration Editor, are used to create user set point variables beyond the PLC's 30 module limit [up to 15 modules per station with a two (2) station maximum]. Q points allow you to access and use the PLC's 168 free (unused) memory locations.

Q points are non-scaled, 32-bit floating point values that are readable and writable over telemetry via DFS radio or network (NIM) protocol. The Q point registers reside in the 49000 range beginning at register 49464 and ending at register 49798. Because Q points are 32-bit floating point values, each point requires two registers; each Q point begins on an even register (e.g., Q point number 50 resides at registers 49562 and 49563).

Q points do not have to be configured in I/O Builder. The 168 Q points are automatically added to the Logical Memory Map when a PLC project is created. Each point is given a label that begins with the letter Q (e.g., Q1, Q2, Q3, etc.). Additionally, they do not have to be mapped from the Logical Memory Map into the DFS Radio Map. Q points are mapped internally and respond to specially-formatted messages sent from HT3. Like other I/O, Q points can be used in ladders and in custom screens.

It is important to note that Q points are not designed to be used as status points. In HT3, they are not polled as often as “normal” I/O; doing so would negatively impact the polling loop. However, as a set point variable, the control is acted on immediately.

To add Q points to an existing PLC project, you must change the project type to RDP and then change it back to PLC. If you have I/O in the Logical Memory Map that is mapped to the registers assigned to Q points, they will not be overwritten. The Q points will be placed “around” the taken registers and will always start at an even register number.

Like the reserved set point registers set aside for storing user set point values (discussed on page 4), Q Point values are automatically saved to the PLC's flash memory every 30 seconds and also during a controlled shutdown. These stored values are loaded when the process resumes.

PROGRAMMED WITH LADDER LOGIC

The PLC is programmed using Logic Builder, a user-friendly application that enables you to construct "ladder logic"-style programs that manage complex control functions. Ladder logic is a graphical (symbols and text) language that is used to plan, maintain and control industrial systems. Each rung of the ladder (hence the name - ladder logic) is used to control a single output or multiple outputs by using branches.

The results of these graphical programs are logical points and auto controls that are continuously scanned by the system. The speed of the scanning process enables you to have the most up-to-date information, which, in turn, allows you to react to situations quickly and efficiently.

In traditional ladder logic, the values that flow along rungs and branches are strictly logical, 0 or 1. DFS' Logic Builder provides the extra flexibility of allowing rungs and branches to hold numeric values (for example, the results of math operations, such as ADD and MAXIMUM, and inputs from analog points).

After a ladder logic program has been built and installed, its logical points can be used in custom screens (graphical representations of your process control system) that can be used to remotely monitor and control your system.

Additionally, you can use PMT’s Simulator to test how your ladder logic will behave under specific conditions. If you are connected to the PLC and receiving real data, Simulator can be used to troubleshoot your logic.

MONITOR AND CONTROL VIA CUSTOM SCREENS

Graphical representations of your process control system can be created using the Screen Builder application. Building a screen – using text, images, objects, and animation – and linking the screen's components to real or logical I/O, enable you to get a real-time view of your operation as well as control processes from a remote location.

For example, you can build a screen that shows the flow of a pump or the level in a well, and then use the device's mapped address (register) to link it to the real field hardware. This linking lets you create a virtual picture of how the equipment is operating; the screen mimics the activity of the equipment. It also enables you to control devices from a remote location. For example, a screen with an On/Off button that is linked to a pump motor can be used to start or stop a pump.

PMT's Simulator can be used to test how your custom screens will behave under specific conditions.

INSTALLING PMT

The Process Management Toolkit is compiled to run in a Linux and Windows XP (or greater) operating system environment and requires the Java 2 Platform, Standard Edition (J2SE) Java Runtime Environment (JRE) 1.5.0 or later.

Java 1.5.0 is included in the installer and will be installed on your computer if you have an older version of Java. Note that installation of this version of Java will not over-write or otherwise affect any version of Java already installed on your computer.

There is no need to uninstall any previous versions of PMT.

1. To launch the setup wizard, double click the file named PMT2_windows_1_0.exe, which was provided with your PLC.
2. Follow the onscreen instructions to install the program in the default directory C:\Program Files\PMT2. During installation, the wizard will place shortcuts in the start menu and on the desktop.
3. Click Finish when prompted to exit the setup wizard.

Images used in custom screens are stored in a folder named Images that can be found in the main PMT2 folder. The images are the same ones used in the HT3 system. You can use an FTP program to download the images from your HT3 system into this folder. You can also place your own custom screen images, for example maps or photos of your facility, into this folder.

USER INTERFACE

The first time you start PMT, you are presented with a blank window like that shown below. Once you've built custom screens, you can select a custom screen that you want to display each time the project is opened (select **Settings** from PMT's **File** menu).

The PMT window is made up of four parts:

- The **menu and button toolbars** are at the top of the window.
- The **tools panel** is on the left side of the window. It contains supplemental utilities used in creating and testing projects and interfacing with the PLC. These tools include a point picker, a simulator, inspectors for Logic Builder and Screen Builder, and a list of the most recently opened projects. Each tool can be maximized or minimized (opened or closed) by clicking the double arrows located to the right of the tool's name.
- The **status bar** is at the bottom of the window. This area is used to display error and informational messages. Status icons (for example, the network connection icon) also appear in this area.
- The main **building and viewing area** is on the right side of the window. Custom screens are displayed in this area. PMT's building utilities (for example, I/O Builder and Logic Builder) also open in this area.

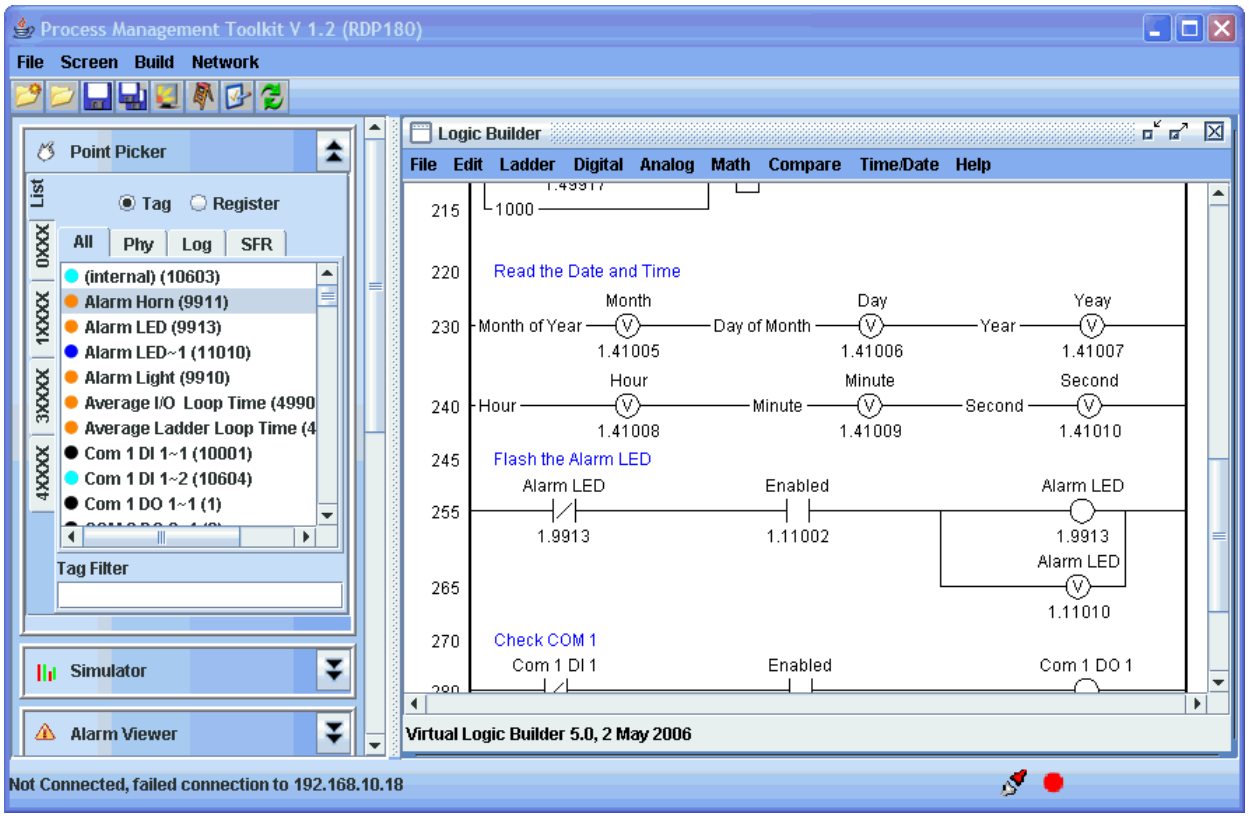


Figure 2-1, PMT User Interface

MENUS

At the top of the window, are the File, Screen, Build, and Network menus.

The **File** menu features the following commands:

- **New** – Create a new project.
- **Open** – Open an existing project.
- **Save** – Save the current project.
- **Save As** – Save the current project with a new file name.
- **Settings** – Specify, view, or edit basic settings including mode (RDP or PLC), target IP address, password, full screen on startup option, and default custom screen to display when PMT is opened. For a PLC that will not be a device on the network, leave the IP at the default setting of 192.168.1.10.
- **Retrieve** – Download configurations from a PLC. The IP address of the PLC that the configurations are being downloaded from must be specified using the Settings command (see above).
- **Install** – Transfer or upload I/O configurations to the PLC. The IP address of the PLC that the configurations are being installed to must be specified using the Settings command (see above).
- **Exit** – Exit the Process Management Toolkit.

The **Screen** menu shows only the word "blank" until you have created at least one custom screen. As you build and save screens, the screen names will be added to this menu. They will be listed in alphabetical order. When you select a screen from this menu, it will open in Screen Viewer. Screen Viewer opens in a new window.

The **Build** menu features commands for opening I/O Builder, Screen Builder, and Logic Builder. All of these utilities open in a new window, which allows you to work on different parts of a system configuration at the same time.

- **I/O Builder** is where you add and configure devices and I/O, including DFS modules and Modbus I/O devices, such as the RIO032 and RIO128.
- **Mapper** is used to map all physical I/O to the correct register range in the PLC's Logical Memory Map. By mapping all points, both DFS-type points and Modbus-type points, to a common map using a common format, you are able to easily create logic for devices that use different communication protocols. Additionally, if the PLC is to be polled by an HT3 system using a DFS protocol (RIM protocol – radio link; NIM RTU protocol – network interface), Mapper is used to map the Logical Memory Map registers into up to 30 DFS modules in the Radio Map [up to 15 modules per station with a two (2) station maximum].
- With **Screen Builder** you create user interfaces to the PLC and the devices connected to it. These screens can show you the status of a device (is the pump running or off? what is the tank's level?). They can also be used to control a device (click a button to turn a pump on or off; enter a value in an input field to force the tank to the specified level).
- **Logic Builder** is used to create the logical, ladder-based programs that run on the PLC.

The **Network** menu features commands for connecting to and disconnecting from the PLC over the network. It also includes Datatap, which is used to test and troubleshoot the PLC's serial communications and monitor the DFS radio or module bus.

The **Help** menu features commands for opening the PMT User Manual (**Browse Help**) and for creating a print-friendly list of all of your project's components, including network settings, devices, modules, physical and logical registers, and logical and radio mapping (**Browse I/O Mapping**).

BUTTON TOOLBAR

Directly below the menu bar are buttons for performing tasks such as creating a new project, saving a new project, and launching Screen Builder. You can use these buttons instead of the menu commands to quickly perform a task. When you place your mouse over a button, a tool tip that describes the button's function appears.

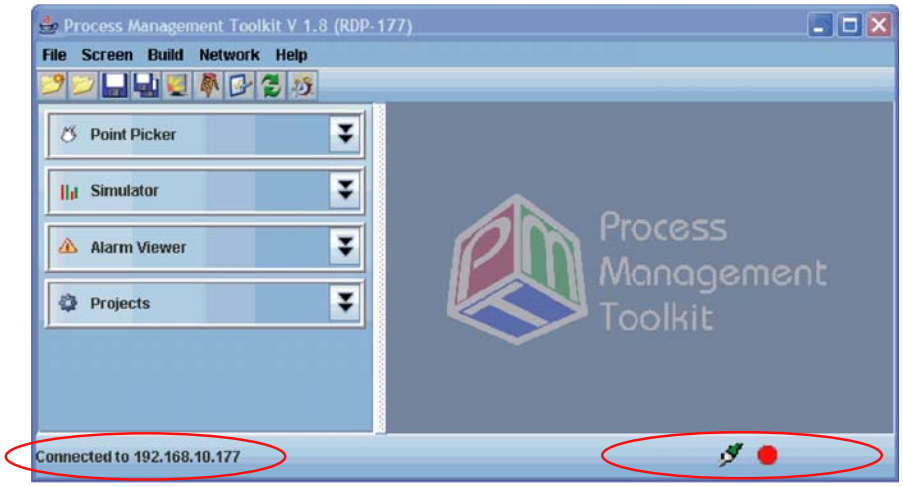
Icon	Tool Tip	Function
	New Project	Create a new project
	Open Project	Open an existing project
	Save Project	Save the current project
	Save Project As	Save the current project with a user supplied file name
	Screen Builder	Open Screen Builder
	Logic Builder	Open Logic Builder
	I/O Builder	Open I/O Builder
	Mapper	Open Mapper
	Alarm Builder	Open Alarm Builder

STATUS BAR

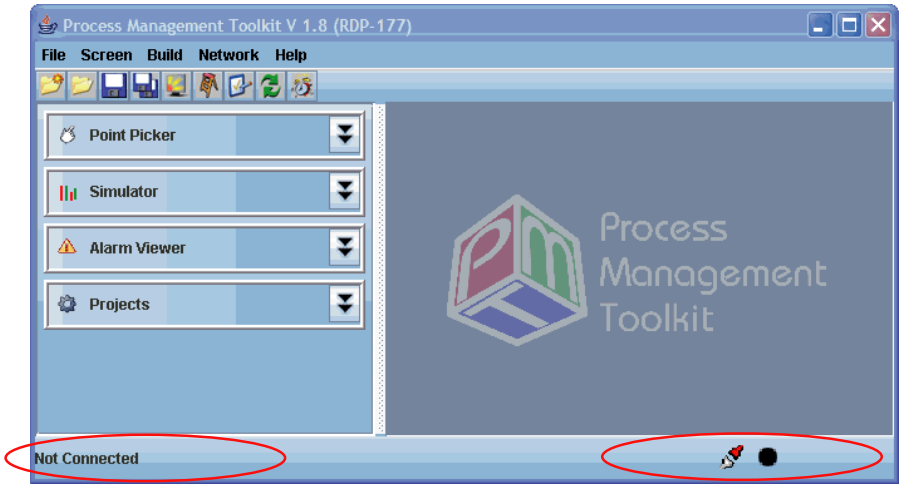
The status bar displays error and informational messages. The network connection status and alarm status also appear in this area (on the right side of the status bar).

In the image below, the network connection status and alarm icons are circled.

- The network connection status icon is green when you are connected to the device.
- The alarm status icon is red when there is an active alarm that has not been acknowledged. The icon is black if there are no active alarm or if all active alarms have been acknowledged. See “Chapter 6: Alarms” for information on creating and configuring alarms.



The image below shows how the status bar appears when you are not connected. Note that the network connection status icon is red and the two halves of the icon are separated to indicate a broken connection; the alarm status icon is black.



CREATING A PROJECT

All of the files associated with the PLC (configurations, ladder logic programs, custom screens) are stored in a single project folder. This folder can reside in any location on your computer.

Before you create your first project, you may want to create a folder on your computer where you can store all of your projects. For example, create a folder named "PLC Project Files" in your "My Documents" folder.

To create a project:

1. Open the Process Management Toolkit software.
2. Select **New** from the **File** menu.
3. Select **Save As** from the **File** menu.
4. Use the **Look In** drop-down list to browse your hard drive and select a folder to save the project files in.
5. Type a name for the project in the **File Name** box.
6. Click **Save PMT Project As**. A folder with the name provided in step 5, above, is created. This folder will hold all of the files associated with this project, including logic.rdp, which is the main project file.

You will notice that the project folder also contains several .CSV files. You can view and print the .CSV files, but you should *not* edit them outside of the Project Management Toolkit. Viewing the .CSV files in a spreadsheet program can be useful if you need to see a list of all of the physical I/O and logical I/O (I/O generated by ladder logic), and how they have been mapped into the Logical Memory Map.

After you have created the project, you must select its mode (PLC or RDP). You can do this through the **Settings** dialog box (select **Settings** from the **File** menu).

IMPORTANT: You should not edit any of these .CSV files outside of the Process Management Toolkit. These files are for informational purposes only.

At any time during the development of your project, you can create a print-friendly list of all of your project's components, including network settings, devices, modules, physical and logical registers, and logical and radio mapping by choosing **Browse I/O Mapping** from the **Help** menu.

CONVERTING A PMT 1.0 PROJECT

When you open a PMT 1.0 project in PMT 2.0, a backup copy of the project is placed in the project folder in a subfolder named "bak." The project is then converted to a format compatible with PMT 2.0.

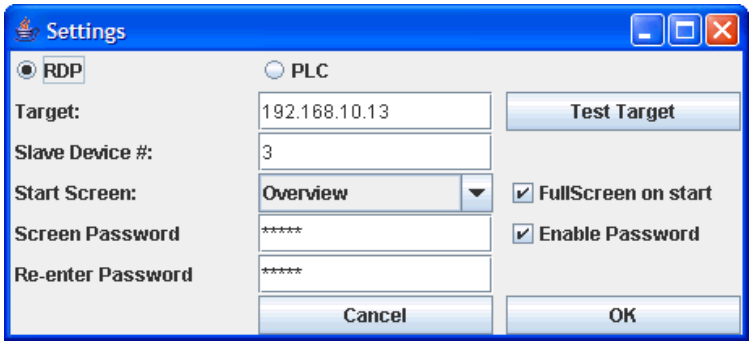
If you need to open the project in PMT 1.0 after it has been converted, you must first delete the file named settings.dfs, which can be found in the corresponding project folder. Once you've deleted this file, you can open the project in PMT 1.0 by browsing to the backup copy of the project.

IMPORTANT: A project created in PMT 2.0 must *not* be opened using PMT 1.0.

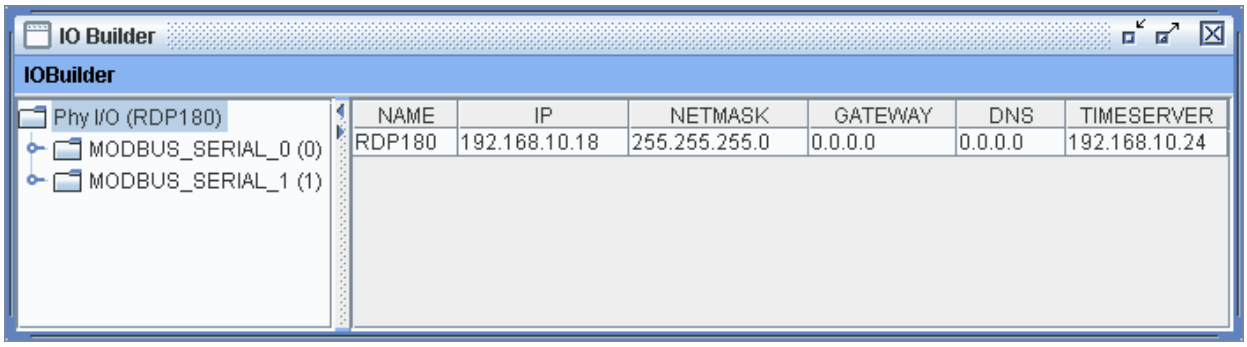
PROJECT SETTINGS

The basic settings for each project are defined in the Settings dialog box (select **Settings** from the **File** menu) and the first node in I/O Builder.

The Settings dialog is used to define the project's mode (PLC or RDP), target IP (used while programming and configuring), serial slave device number, start up screen, and optional password.



The first node in I/O Builder is used to define the project's network settings.



PROJECT MODE

Before you begin configuring and mapping a PLC's I/O, you must select the appropriate mode for the PMT project – PLC or RDP.

Open the **Settings** dialog. Select PLC when configuring a PLC033; select RDP when configuring an RDP033 or an RDP180.

SERIAL SLAVE DEVICE NUMBER

When the PLC or RDP is being used as a Modbus serial slave device, you must specify its Modbus device number (1-254) in the Settings dialog box. This allows the PLC/RDP to operate as a serial slave device on RS-485 or radio networks where there is more than one slave device. If no device number is specified, the PLC/RDP will respond to any serial messaging.

START UP SCREEN

Once you've built custom screens for your project, you can select a custom screen that you want to display each time the project is opened.

Open the **Settings** dialog. Select a screen name from the drop-down list or select BLANK if you don't want a screen to open each time you open the project.

FULL SCREEN ON START (OPTIONAL)

This setting causes Screen Viewer to open in full-screen mode when PMT is started. The screen selected in Start Up Screen (above) is the screen that will be opened.

Using this setting in conjunction with a password (see below) prevents unauthorized users from making changes to any of the project's settings, configurations, and programming. Unless a user has the password, they will only be able to view the selected start up screen.

See "Full Screen Mode" on page 19 for more information.

PASSWORD (OPTIONAL)

This setting enables you to require a password to exit Screen Viewer's full-screen mode.

This is useful, for example, if you want to display a specific custom screen on a flat-screen panel in your plant, but you want to restrict access to the rest of the PMT project. This helps prevent unauthorized users from making changes to configurations and the ladder logic program. Once the screen is in full screen mode, the only way to access the remainder of the applications is to exit full screen view using the assigned password.

Passwords can contain numbers only; letters and special characters are not accepted.

To set a password, open the **Settings** dialog. Select the **Enable Password** option and enter the desired password in the **Screen Password** and **Re-enter Password** boxes.

NETWORK SETTINGS

The PLC must be on a network – either a local area network (LAN) or a closed network – to be configured and programmed.

When a PLC leaves the factory, it is set to a default IP address of 192.168.1.10. PLCs that will be using a radio link to communicate can be left at this default setting. However, if a PLC is going to be a network device, your network administrator must assign it a unique network IP address. The PLC is configured with this assigned "*destination*" IP address via I/O Builder. If the PLC isn't going to be a network device, leave the destination IP at the factory default setting.

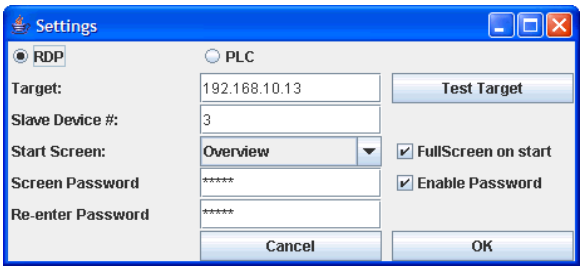
A second IP setting – the Target IP (select **Settings** from the Process Management Toolkit's **File** menu) – is only used when installing and retrieving configurations. The Target IP is the PLC's *current* IP. For a PLC that has just arrived from the factory, the Target IP is 192.168.1.10 (the factory default). For a PLC that has been configured, the Target IP will be the IP specified in I/O Builder. Note that if the PLC hasn't been used as a network device, then the destination IP (the address specified in I/O Builder) is the factory default 192.168.1.10 *unless* this was changed by someone configuring the PLC.

Normally, the IP addresses specified in **Settings** and I/O Builder will be identical. However, when you are first setting up a network-bound PLC, the IP addresses will be different. In I/O Builder, you will enter the IP that you network administrator assigned to the PLC (its *"destination"* IP). In the **Settings** dialog, you will enter the PLC's current IP – the factory default 192.168.1.10. This allows you to configure the PLC with its *destination* IP address while communicating with the PLC using its *current* (Target) IP address.

Another scenario where the Target and Settings IP could be different is if you needed to move the PLC to another IP address on your network. You would set the Target IP in Settings equal to the PLC's current address and then provide the PLC with its new destination IP address via I/O Builder.

TARGET IP

The Target IP is used only when installing and retrieving PLC configurations. To enter the PLC's Target IP, select **Settings** from the **File** menu. The **Settings** dialog box opens.



Many times a PLC that will be a network device is initially configured and programmed on a test bench. In this instance, you create a closed network by connecting the PLC directly to a desktop or laptop computer using a network crossover cable. You would then change the network settings for your computer, so that it is on the same network as the PLC.

For example, if the PLC was still at the factory default IP address (192.168.1.10), the computer would also have to be on the 192.168.1 network (e.g., set the computer's IP address to 192.168.1.101). The PLC's Target IP would be 192.168.1.10.

A closed network is also used when installing and retrieving configurations for a PLC that will never be on the network (for example, a PLC in a radio RTU). In this case, the target IP address and the IP addressed specified in I/O Builder (below) would be the same (192.168.1.10).

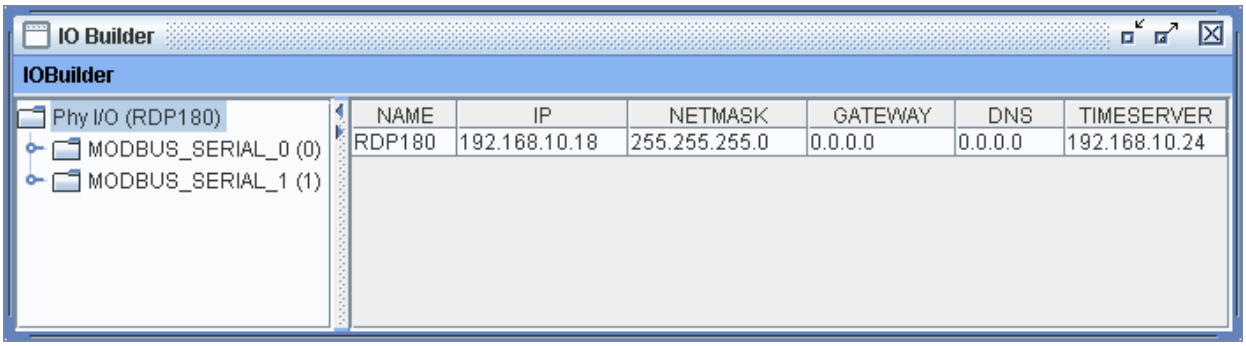
If you are going to be configuring and programming the PLC over the local area network, no changes need to be made to your computer. Simply enter the PLC's current IP address in the **Settings** dialog to connect to it.

Test Connection

1. Click the **Test Target** button to ping the PLC at the IP entered in the **Target** box.
2. After clicking **Test Target**, a window appears with the message "Connecting to: xxx.xxx.xxx.xxx" (where xxx.xxx.xxx.xxx represents the IP address entered in the **Target** box).
3. If a connection is made, the message "Login successful: xxx.xxx.xxx.xxx" followed by the word "Bye" will be displayed. Otherwise the message "Could not connect to: xxx.xxx.xxx.xxx" will be displayed.

DESTINATION IP

To configure the PLC's network communication settings, open I/O Builder (select **I/O Builder** from the **Build** menu), and select the top node of the tree (the one beginning with Phy I/O).



Check with your network administrator if you are unsure of any of the settings discussed below.

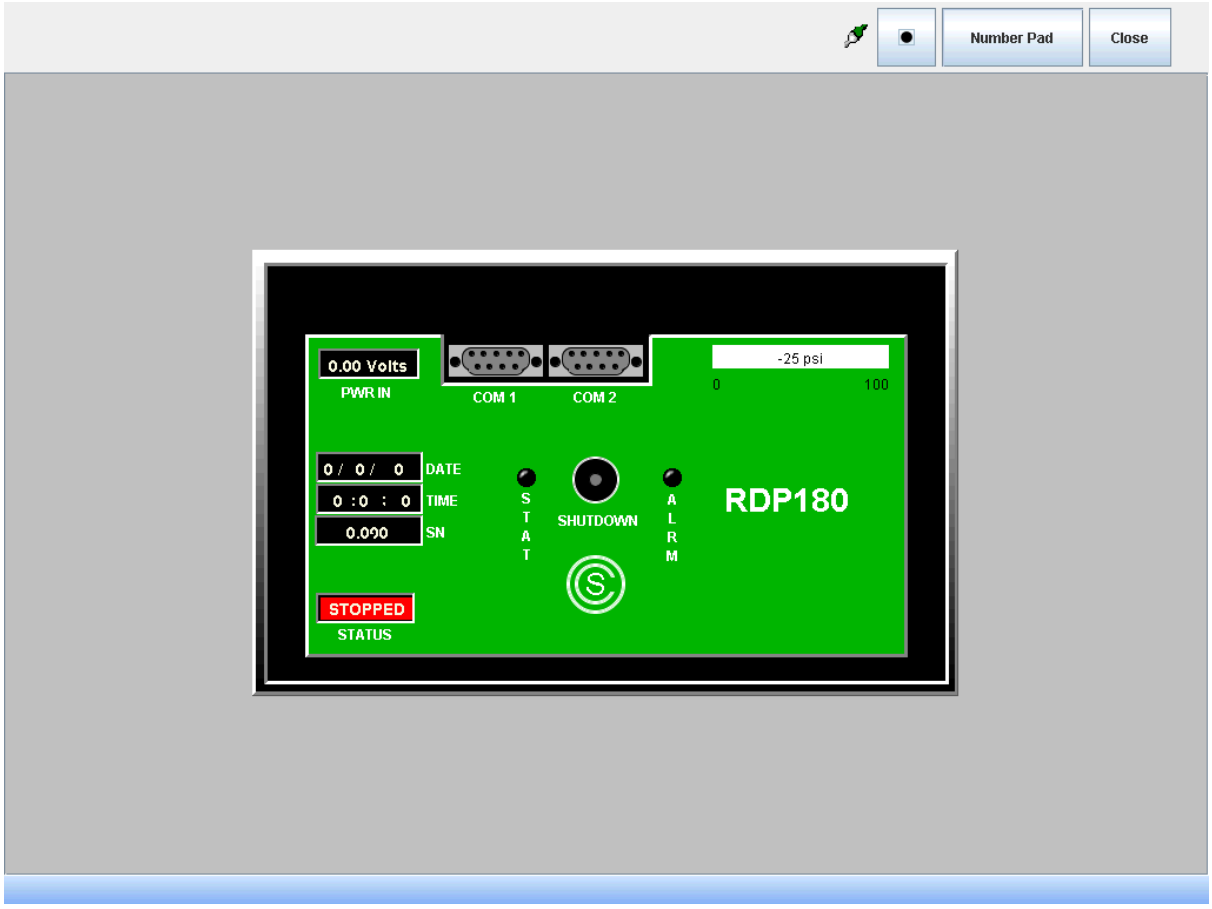
- **Name** – Enter the user-defined name for the PLC. Up to 20 characters can be entered in this field. Valid characters are the letters A-Z, the numbers 0-9, and the characters. (period), / (forward slash), _ (underscore), @ (at symbol), and $ (dollar sign). The name entered here will replace the word "untitled" at the top of the tree in I/O Builder. The name entered here uniquely identifies the PLC. This can be useful when editing a configuration (you know exactly what PLC you're working on).
- **IP** – Enter the IP address assigned to the PLC (i.e., its *destination* IP). If the PLC will be installed in a non-network RTU (radio-only RTU), leave the IP address at the default 192.168.1.10.
- **Netmask** – Enter the subnet mask of the network that the PLC is on. The default setting of 255.255.255.0 should only be changed under the direction of your network administrator.
- **Gateway** – Enter the IP address of the router that is used to connect the PLC to all other networks in the system.
- **DNS** – Enter the IP address of the Domain Name System (DNS) server used by your network.
- **Timeserver** – Enter the IP address of your timeserver. The PLC uses the specified timeserver to synchronize its system time. The timeserver can be on your own network, or can be a server on the Internet. (**NOTE**: Timeserver uses the Unix/Linux RDATE command.)

IMPORTANT: Do not configure a timeserver if the PLC is being polled by an HT3 server via DFS RIM (radio) or NIM RTU (network) protocol. When the PLC is polled using either of these protocols, a network time sync message is sent out once a polling loop. It is the first message sent at the beginning of every loop. Configuring a timeserver in this instance could adversely affect any time-related functions in your ladder. The timeserver and the HT3 server could constantly be setting the PLC's time to conflicting values causing unexpected and unintended behavior with time and date functions in the ladder. Note that a PLC033 will sync its time once an hour to match the time included in the time sync message if it determines that its time is out of sync with the HT3 server.

FULL SCREEN MODE

I/O Builder, Mapper, and Screen Viewer have a full screen option. When Full Screen is selected, the tool's workspace takes up the entire screen. With a larger workspace, you can see more information without scrolling left and right, or up and down.

With Screen Viewer, you can also require a password in order to exit full-screen mode. This enables you to restrict access to the rest of the PMT project; unauthorized users are prevented from making changes to configurations and the ladder logic program. Once the screen is in full screen mode, the only way to access the remainder of the applications – except Alarm Viewer – is to exit full screen view using the assigned password.

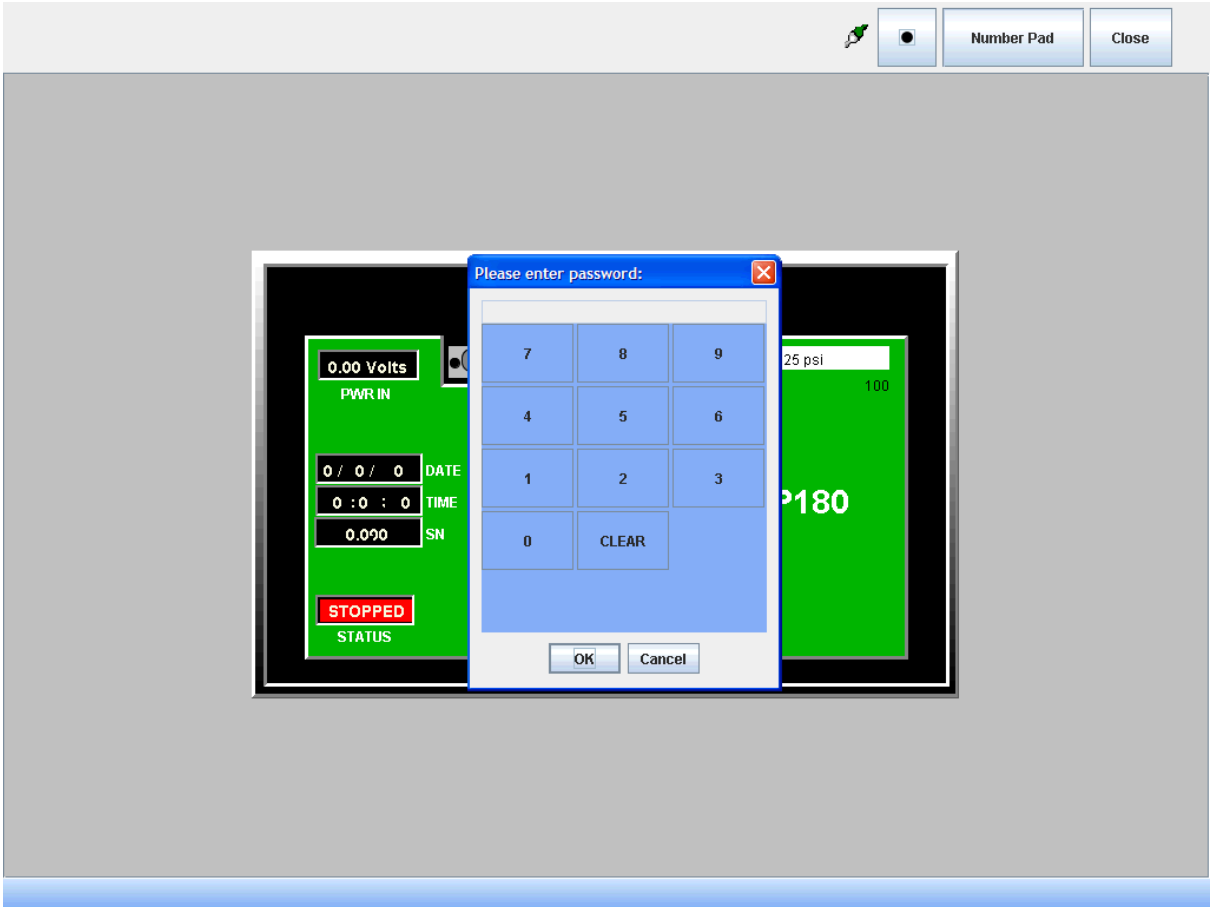


At the top of the full-screen mode window is an icon for network connection status and an icon button for alarm status. These are similar to the icons that appear at the bottom of the PMT window during normal-view mode.

One difference is that the alarm status icon is a button that opens Alarm Viewer. This is a fully-functional version of Alarm Viewer; alarms can be sorted by tag name or status and can be acknowledged by clicking the alarm.

Next to the alarm status icon button is the Number Pad button. Clicking this button switches the application to number pad mode. This mode is useful for screens being viewed on a touch screen monitor that doesn't feature a keyboard and mouse. When this mode is enabled, touching a screen object such as

an analog control will open a numeric keypad on the screen. Using this keypad you can enter the value you want to control the object to. This keypad would also be used if the password option had been enabled.



EDITING BASICS

EDITING A FIELD

To edit a field, double click inside the box that contains the text or data that you want to change. Press the DELETE key to clear the current contents and type in the new text/data; or, simply begin typing while the field's contents are highlighted. Some fields have drop-down lists from which you can select the desired setting.

Any time a field is edited, the background of the line containing that field turns gold to alert you that a change has occurred and that the changes should be saved. Once the project file has been saved, the line returns to its normal color.

COPY AND PASTE

You can copy and paste individual cells, ranges of cells, and entire rows and columns.

- Use CTRL+C to copy. Or, right click the cell or range of cells and select Copy from the pop-up menu.
- Use CTRL+V to paste. Or, right click the cell or range of cells and select Paste from the pop-up menu.

To copy and paste:

1. Select what you want to copy and press CTRL+C.
2. Place the cursor where you want to paste the cells and press CTRL+V.

NOTES:

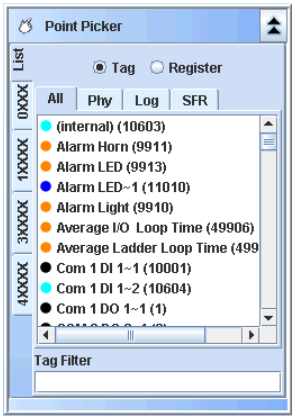
- If you are copying an entire column or row, you must select an entire column or row to paste the data into.
- If you are copying a range of cells, the area that you are pasting into must have the same "size and shape" (e.g., if the copy selection is two columns wide and three rows high, the area that you paste the data into must also be two columns wide and three rows high).
- The contents of one cell can be copied and pasted into a range of cells. After copying the cell, select the range of cells to paste the data into and press CTRL+V.

DELETING ROWS

From the tree, select the item (device or point) to be deleted. Right click and select **Delete** from the pop-up menu.

POINT PICKER BASICS

Point Picker is a utility used to select registers for use with Simulator, Logic Builder, and Screen Builder. Point Picker lists all of the used (reserved) registers. This includes registers assigned to physical points, logical points, and special function registers. Registers in Point Picker are categorized by type. Each type is represented by a colored dot.



- Black – physical points
- Blue – logical points
- Orange – special functions registers (SFRs)
- Cyan (light bluish green) – system variables

OPENING, CLOSING, AND MOVING POINT PICKER

When PMT is opened, Point Picker is opened at the top of the tools panel (left panel of the PMT window).

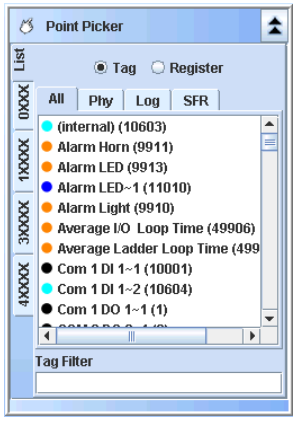
To close Point Picker, click the two upward facing arrows on the top right of the Point Picker window; to open Point Picker, click the two downward facing arrows.

You can also move Point Picker – and the other tools – up and down in the tools panel to rearrange their order.

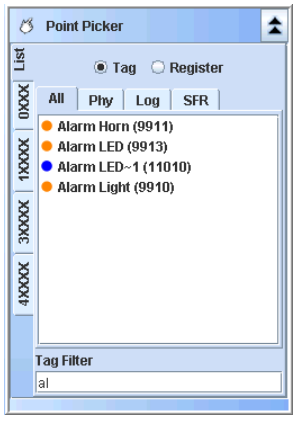
1. Click Point Picker's title bar. The cursor changes from an arrow to a box.
2. Continue holding down the mouse button and drag Point Picker up or down.
3. Release the mouse button when you have the tool in the desired location.

SORTING AND FILTERING OPTIONS

SORT BY TAG NAME

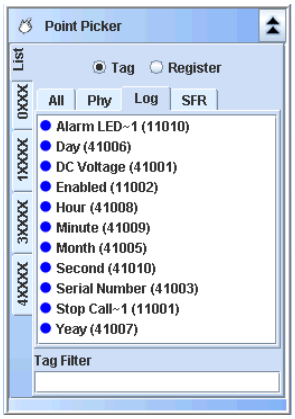


Select the **List** tab and then choose the **Tag** option to see a list of points sorted by their tag name (user-defined name assigned to them in I/O Builder or Logic Builder; or system-assigned name given to special function registers or logic builder functions, e.g., Multiply and Latch).



Filter by Tag Name

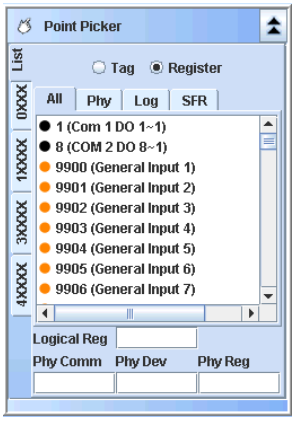
Points sorted by tag name can be filtered by entering text in the **Tag Filter** box. As text is typed into this field, the list of registers is narrowed. The example at left shows a list of all registers whose tag name begins with the letters "al".



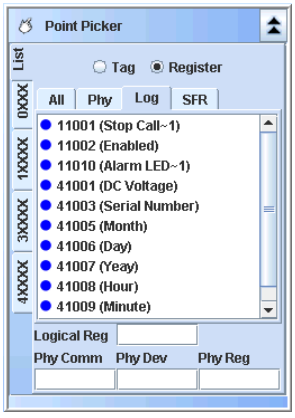
Filter by Type

Points sorted by tag name can also be filtered by their type. By selecting the **Phy**, **Log**, or **SFR** tab, you can see a list of physical, logical, or special function registers. The example at left shows a listing of all logical points sorted by tag name.

SORT BY REGISTER



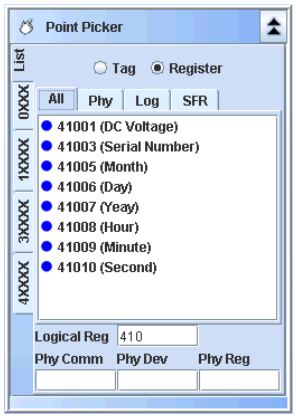
Select the **List** tab and then choose the **Register** option to see a list of all points sorted by their register number.



Sort by Register; Filter by Type

Points sorted by register can also be filtered by their type. By selecting the **Phy**, **Log**, or **SFR** tab, you can see a list of physical, logical, or special function registers. The example at left shows a listing of all logical points sorted by register number.

(continued on next page….)



Sort by Register; Filter by Logical Register

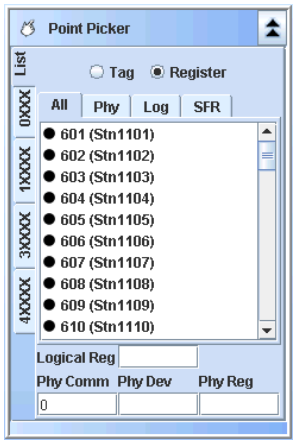
An additional way to filter and search for records that are sorted by register is to select the **All** tab and enter information in the **Logical Reg** (logical register) box.

As text is typed into the **Logical Reg** field, the list of registers is narrowed. For example, typing 410 in this field would show you a list of all logical registers in the 410XX range.

Sort by Register; Filter by Physical Characteristics

The physical communication driver (**Phy Comm**), physical device (**Phy Dev**), and physical register (**Phy Reg**) fields can be used in conjunction with each other to search for a logical register using characteristics of the corresponding physical register (its driver number, device number, and register number). As information is typed in each of the boxes, the list narrows to match the filter(s).

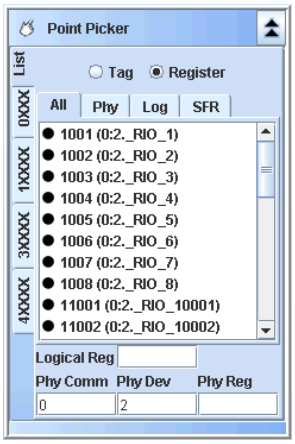
This can be useful if you used an offset when you mapped your physical I/O registers into the logical memory map. Because there isn't a one-to-one relationship when mapping is done this way, it may be difficult to remember the logical register that the physical register was mapped to. For example, a digital input with a physical register of 10001 that was mapped to logical register 105001 (offset equal to 500).



Sort by Register; Filter by Driver Number

Select the **All** tab and enter a driver number (this is the number displayed in parentheses next to the DFS BUS driver and the Modbus Serial driver) in the **Phy Comm** field to see a list of all the physical I/O registers configured for that driver sorted by their corresponding logical register.

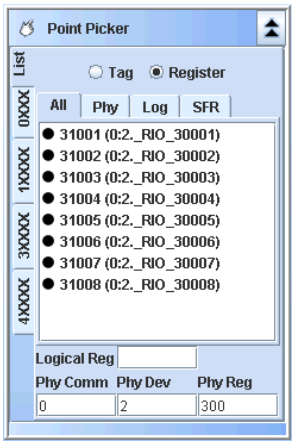
The example at left shows a list of all registers for driver 0.



Sort by Register; Filter by Device Number

Leave the driver number in the Phy Comm field and enter a device number in the Phy Dev field to can see a list of all physical registers configured for that driver and device sorted by their corresponding logical register.

The example at left shows a list of all registers for driver 0 and device number 2.



Sort by Register; Filter by Register Number

Leave the driver number in the Phy Comm field and the device number in the Phy Dev field, and enter a full or partial register number in the Phy Reg field (for example, 300 for anything in the 300XX range) to see a list of all physical registers configured for that driver and device that fall in the range entered in the Phy Reg field. The list will be sorted by the corresponding logical register.

The example at left shows a list of all registers for driver 0 and device number 2 that are in the 300XX range.

SORT BY I/O TYPE

Registers can be sorted by I/O type (digital output, digital input, analog output, analog input) by selecting the tab along the left edge of Point Picker that corresponds to the beginning register range for that I/O type.

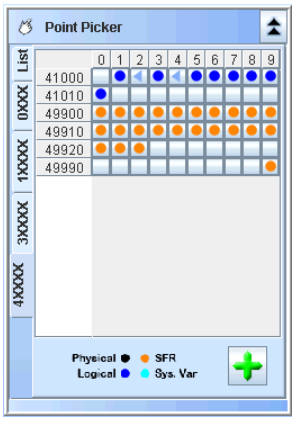
- Select **0XXX** to see all digital outputs (coils)
- Select **1XXXX** to see all digital inputs (discrete inputs)
- Select **3XXXX** to see all analog inputs (input registers)
- Select **4XXXX** to see all analog outputs (holding registers)

In this mode, the points are displayed in a tabular format.

The cell of each used (reserved) register contains a colored dot that indicates the register's type (physical, logical, special function). A legend at the bottom of the window lists the colors associated with each type.

A cell that contains an arrow indicates that the register is reserved as the lower half of a 32-bit number. The cell to the left of the arrow (the direction the arrow is pointing) is the upper half of the

number. These 32-bit numbers are *not* user configurable; they are system variables created by Logic Builder.



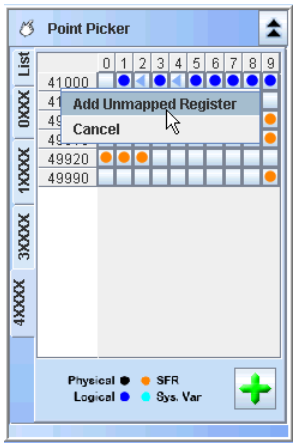
ADDING / REMOVING UNMAPPED REGISTERS

Using the tabular view in the point picker, you can set aside the logical registers you will need in your ladder. For example, you can reserve registers for all of the digital and analog Virtual Out objects needed in your ladder. When you start building your ladder, you can use the point picker to assign a register to a logical object instead of typing in a register.

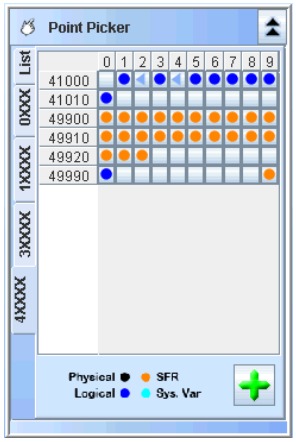
Note that these are “unmapped” registers. If you want to query them from an HT3 system using the DFS radio or network protocol, you must use Mapper to map them into the appropriate module for their type.

ADDING A REGISTER

1. Select the tab on the left side of the point picker for the type of register you want to add (0XXX, 1XXXX, 3XXXX, or 4XXXX).
2. Choose an unused register (one without a dot in it) and click.
3. Select **Add Unmapped Register** from the pop-up menu.



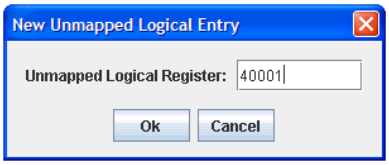
4. The cell is filled with a blue dot to indicate it is reserved for a logical point. You can now use the point picker to assign this register to a logical (virtual) object in your ladder logic program.



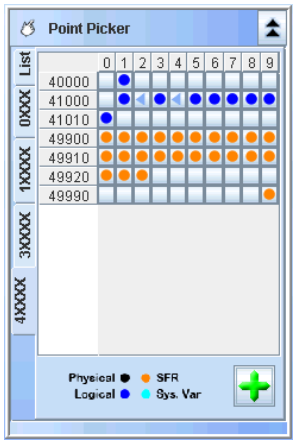
ADDING A REGISTER IN A NEW RANGE

The tabular view of point picker only displays the register ranges that have been used. However, it is possible to reserve a register in one of the hidden ranges.

1. Select the tab on the left side of the point picker for the type of register you want to add (0XXX, 1XXXX, 3XXXX, or 4XXXX).
2. Click the green plus (+) button that appears at the bottom right corner of the point picker window.
3. In the **New Unmapped Logical Entry** box, type the number of the register you want to reserve and click Ok.

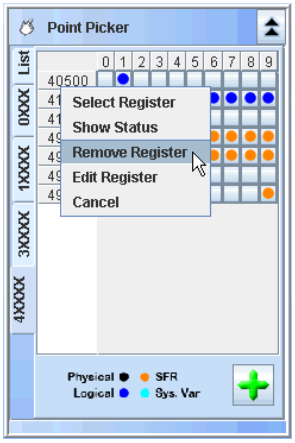


4. A blue dot appears in the cell representing the selected register.



REMOVING A REGISTER

You can free up a register you reserved by clicking the register and selecting **Remove Register** from the pop-up menu.



SIMULATOR BASICS

PMT's Simulator can be used in two ways depending on whether or not you are connected to the PLC.

- When you are disconnected from the PLC, you can simulate system activity by setting simulate values.
- When you are connected to the PLC, this utility behaves as an animator showing the real values coming from the PLC and allowing you to send control messages to the PLC.

You can use PMT's Simulator to test how your ladder logic or custom screens will behave under specific conditions.

OPENING, CLOSING, AND MOVING SIMULATOR

When PMT is opened, Simulator is opened beneath Point Picker in the tools panel (left panel of the PMT window).

To close Simulator, click the two upward facing arrows on the top right of the Simulator window; to open Simulator, click the two downward facing arrows.

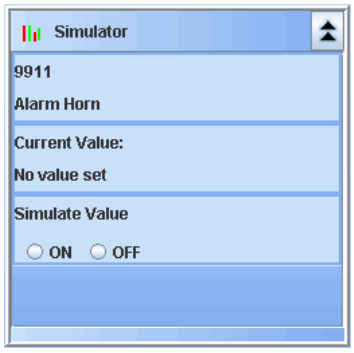
When you first open Simulator, it will display the message "Select Registers in the Point Picker to auto-fill Simulator fields." Once a register has been selected, information including its register number, name, and current value will be displayed.

You can also move Simulator – and the other tools – up and down in the tools panel to rearrange their order.

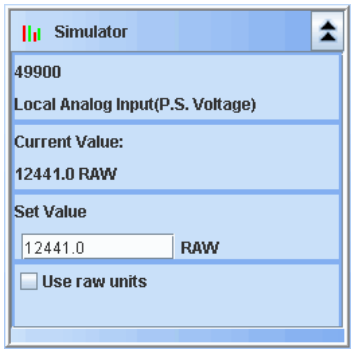
- Click Simulator's title bar. The cursor changes from an arrow to a box.
- Continue holding down the mouse button and drag Simulator up or down.
- Release the mouse button when you have the tool in the desired location.

SETTING AND READING VALUES

1. Select a register using the point picker.
   - If you are not connected to the PLC, the **Current Value** field will display the message "No value set"; the section directly below will be titled **Simulate Value**.
   - If you are connected, the register's current value will be displayed in the **Current Value** field and will be updated whenever the register's value changes; the section directly below will be titled **Set Value**.
2. Set the value using one of the following methods.
   - Digital Register – If a digital-type register was selected, the **Simulate / Set Value** section will display two radio buttons that represent the point's high and low labels. To set the point to either value, simply select the appropriate radio button.



   - Analog Register – If an analog-type point was selected, the **Simulate / Set Value** section will display either an empty box (not connected) or the point's current value (connected). To set or change the point's value, simply type the desired value in the box and press the Enter key.

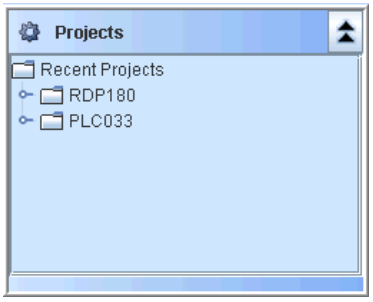


RECENT PROJECTS LIST BASICS

Projects lists up to five of the most recently opened projects. It provides you with a quick way of opening a project; you don't have to browse through your hard drive to locate the project file.

OPENING, CLOSING, AND MOVING THE RECENT PROJECTS LIST

When PMT is opened, Projects is opened below Simulator in the tools panel (left panel of the PMT window).



To close Projects, click the two upward facing arrows on the top right of the Projects window; to open Projects, click the two downward facing arrows.

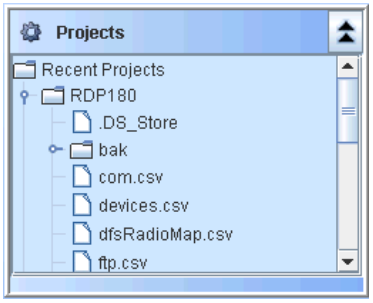
You can also move Projects – and the other tools – up and down in the tools panel to rearrange their order.

1. Click Projects' title bar. The cursor changes from an arrow to a box.
2. Continue holding down the mouse button and drag Projects up or down.
3. Release the mouse button when you have the tool in the desired location.

OPENING A PROJECT

To open a project, simply double click the project's file name in the **Recent Projects** list.

To see a list of all the files contained in the project, expand the project by clicking the key-shaped icon to the left of the project's name.



Note that this listing is for informational purposes only; you can't open any of the .csv files from within PMT.

Chapter 3: CONFIGURING AND MAPPING I/O

OVERVIEW

The PLC is configured over a network using the supplied Process Management Toolkit software. The applications used to accomplish this are:

- **Settings** – This command, which can be found in the File menu, is used to select the mode for the project (PLC033 or RDP), assign a target IP address, and select the custom screen that will be displayed when the project is opened.
- **I/O Builder** – Configure a device's communication and I/O parameters.
- **Mapper** – Map the device's physical I/O into the proper register ranges in the PLC's Logical Memory Map. Additionally, if the PLC is going to be polled by an HT3 system using a DFS protocol (RIM protocol – radio link; NIM RTU protocol – network interface), Mapper is used to map the Logical Memory Map registers into up to 30 DFS modules in the Radio Map [up to 15 modules per station with a two (2) station maximum].

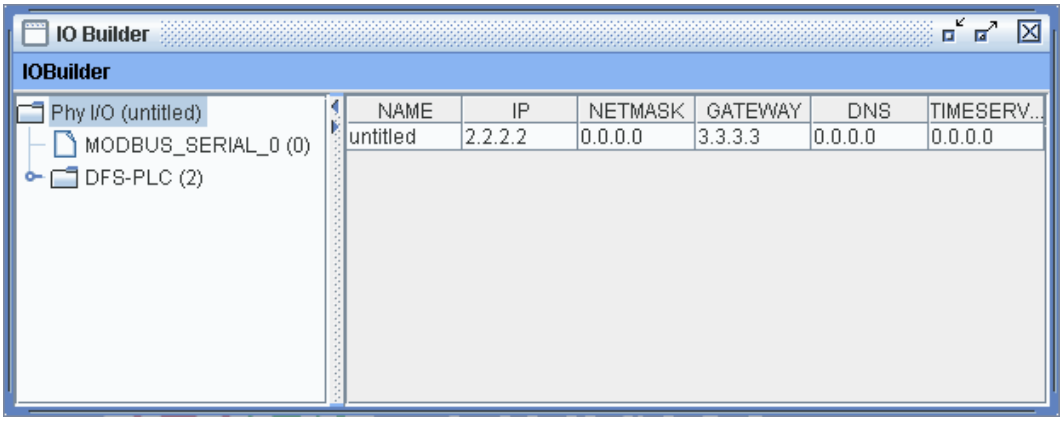
NOTE: It is possible to retrieve configurations from an existing PLC. This is useful if you want to review the PLC's configurations or want to use its configuration at another station. When configurations are retrieved from a PLC, all of the parts of a project are downloaded. This includes I/O configurations, register and DFS radio maps, ladders, and screens. See "Retrieve Information from an Existing PLC" beginning on page 62 for more information.

I/O BUILDER

From the main Process Management Toolkit window, select I/O Builder from the Build menu. I/O Builder opens in a new window.

NOTE: To save changes made in I/O Builder, you must use the commands on the main Process Management Toolkit window. If you close the I/O Builder before saving your changes, the changes are *not* lost. At that point, all you need to do is select **Save** from the **File** menu. If you close the Process Management Toolkit software before saving your changes, you will be prompted to save.

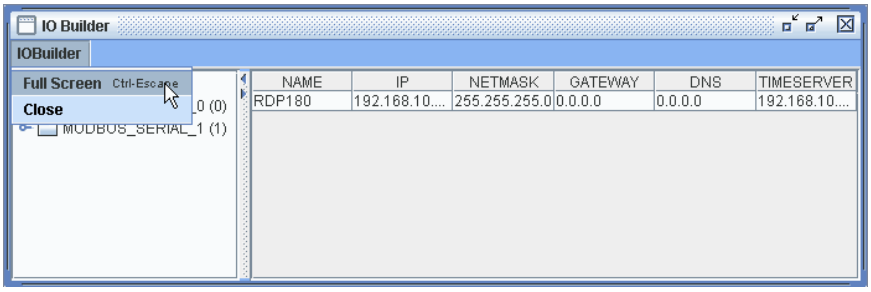
The opening screen of a new project is shown below:



I/O Builder's interface is comprised of two panels. The tree in the left panel is used to navigate through the PLC's parameters. From here you can add/edit communication channels, slave devices, and digital and analog I/O. When you select an item on the tree, its parameters are displayed in the right-side panel. The panel on the right is organized in a table-style format with each row representing one row (or record) in the corresponding .CSV file.

FULL SCREEN MODE

I/O Builder can be viewed in full-screen mode by selecting **Full Screen** from the **I/O Builder** menu.



In full-screen mode, the I/O Builder window fills the entire screen – the tools panel (Point Picker, Recent Projects, Simulator) as well as PMT's menu and icon toolbars are no longer visible. To return to normal mode, click the Close button that appears in the top right corner of I/O Builder. (Note that Mapper can also be viewed in full-screen mode.)

PLC033 SERIAL COMMUNICATION SETTINGS

The PLC033 features three serial ports. COM1 may be used to communicate with external Modbus serial devices (either RS-232 or RS-485). COM2 is used exclusively to communicate with the RTU's Radio Interface Module (RIM). COM3 is used exclusively to communicate with the RTU's local modules.

In this section, you configure the communication settings for the PLC033's COM1 (MODBUS_SERIAL_0) and COM3 (DFS-PLC) serial ports. It is not necessary to configure the settings for COM2.

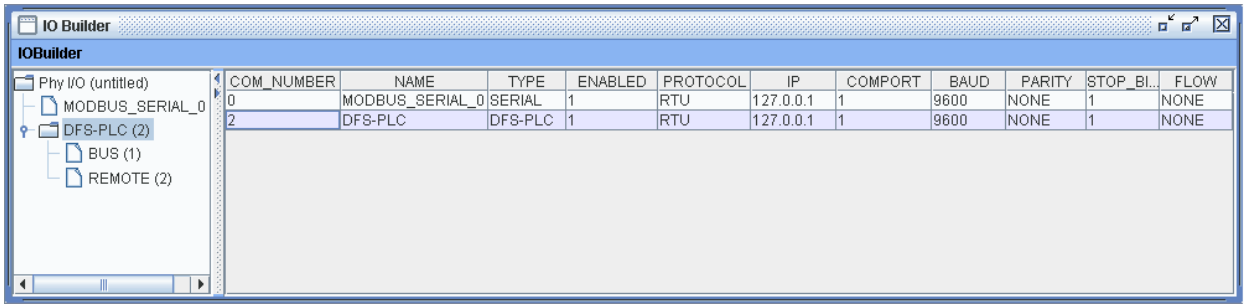
After reading this section, proceed to "Modbus TCP Master Communication Settings" on page 38 if you will be using the PLC033's Ethernet port to communicate with Modbus TCP slave devices. Otherwise, continue to the sections titled "Configure DFS I/O (PLC033 Only)" beginning on page 38 and "Configure Modbus I/O (PLC & RDP)" beginning on page 46.

NOTE: Before beginning this process, verify that PLC mode has been selected in the project's settings. See "Project Settings" beginning on page 15 for more information.

PLC033: COM3 (LOCAL MODULES)

These settings are the ones the PLC033 uses when it communicates with the RTU's local modules – the DFS function modules installed in the RTU. The PLC033 communicates with these function modules over the module bus.

Select DFS-PLC (2) from the tree. The Com Endpoint Table is displayed in the right panel. The parameters for COM3 are listed in the second line of the table.



IMPORTANT: If you are not using the PLC's COM1 port (i.e., only local function modules are being used for I/O; no external Modbus devices are connected to the PLC033's COM1 serial port), assign MODBUS_SERIAL_0 to Comport 0 and set Enabled to 0. This turns off the COM1 driver. Additionally, after making any changes to the serial or network communication settings, you must restart the PLC in order for the changes to take affect.

Only the required settings are listed below. Leave all others at their default values.

- **Enabled** – Select 1 from the drop-down list to indicate this port is being used.
- **Comport** – Select 3 from the drop-down list.

(continued on next page….)

- **Baud** – In most situations, this should be set to 9600. The only exception would be if the PLC033 is communicating with older DFS function modules that have a maximum baud rate of 1200. In that case, select 1200 from the drop-down list.
- **Parity** – Select ODD from the drop-down list.
- **Stop Bits** – Select 2 from the drop-down list.
- **Flow** – Select DFSMOD from the drop-down list.

PLC033: COM1 (Modbus Master or Slave Device)

These settings are the ones the PLC033 uses when communicating with a Modbus master or slave device (for example, a RIO128) that is connected to the PLC033's COM1 serial port. COM1 can be used in either RS-232 mode (Pins 1, 3, 5, 7, and 9) or RS-485 mode (Pins 11, 13, and 15).

When the PLC033 is being used as a Modbus serial slave device, you must also specify its Modbus device number in the project settings. See "Serial Slave Device Number," below.

IMPORTANT: You cannot connect devices to both the RS-232 pins *and* the RS-485 pins. COM1 is a single serial port that can be used in either RS-232 or RS-485 mode.

Review the serial specification of the device you are connecting to the PLC033's serial port to determine the settings you should select.

Serial Slave Device Number

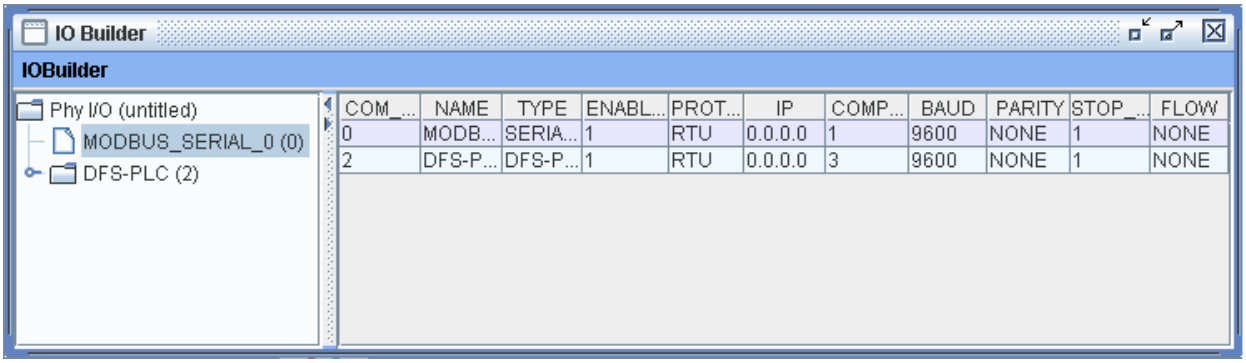
When using the PLC033 as a Modbus serial slave device, you must specify the PLC033's Modbus device number (1-254) in addition to configuring the communication parameters. This allows the PLC033 to operate as a serial slave device on RS-485 or radio networks where there is more than one slave device.

The device number is configured in the Settings dialog box.

- Select Settings from the File menu and enter the desired number in the Slave Device # field.

COM1 Communication Settings

Select MODBUS_SERIAL_0 from the tree. The Com Endpoint Table is displayed in the right panel. The parameters for COM1 are listed in the first line of the table.



IMPORTANT: If you are not using the PLC's COM1 port (i.e., only local function modules are being used for I/O; no external Modbus devices are connected to the PLC033's COM1 serial port), assign MODBUS_SERIAL_0 to Comport 0 and set Enabled to 0. This turns off the COM1 driver. Additionally, after making any changes to the serial or network communication settings, you must restart the PLC in order for the changes to take affect.

Only the settings required are listed below. Leave all others at their default values.

- **Type** – Select SERIAL_SLAVE or SERIAL_MASTER. When SERIAL_SLAVE is selected, the PLC033 will respond over the COM1 port to queries from a Modbus master device. When SERIAL_MASTER is selected, the PLC033 will poll Modbus slave devices over the COM1 port.
- **Enabled** – Select 1 from the drop-down list to indicate this port is being used.
- **Protocol** – Select RTU or ASCII from the drop-down list.
- **Comport** – Select 1 from the drop-down list.
- **Baud** – Select 1200, 9600, 19600, or 38400 from the drop-down list. (Select 38400 for a RIO.)
- **Parity** – Select None, Odd, or Even from the drop-down list. (Select None for a RIO.)
- **Stop Bits** – Select 1, 2, or None from the drop-down list. (Select 1 for a RIO.)
- **Flow** – Select the method of hardware flow control for this device. Choose None, Radio (RTS-CTS), or Bus (CTS) from the drop-down list. (Select None for a RIO; select Bus for an RS-485 device.)

RDP SERIAL COMMUNICATION SETTINGS

The RDP features two independent serial ports: COM1 is an RS-232/RS-485 DCE designed as an interface to a Modbus DTE slave device, such as the RIO128/RIO032; COM2 is an RS-232 DTE port designed as an interface to a Modbus DCE master device (for example, the RDR200). COM1 can be used in either RS-232 or RS-485 mode to communicate with Modbus slave devices.

Although the RDP's serial ports were designed for these specific functions, you are not limited to this configuration. COM1 and COM2 can be used interchangeably to communicate with either DTE or DCE master or slave devices. For example, it is possible to connect a DCE master device to COM1 or a DTE slave I/O device to COM2, or use both ports as interfaces to RS-232 slave devices. Note, however, that null modem (crossed) cables rather than straight cables are required in order to allow two RS-232 like devices to communicate with each other (DTE device connected to RDP's COM2 RS-232 DTE port; DCE device connected to RDP's COM1 RS-232 DCE port).

In this section, you configure the communication settings for the RDP's COM1 and COM2 serial ports. Note that both drivers (MODBUS_SERIAL_0 AND MODBUS_SERIAL_1) can be used for either of the RDP's serial ports (COM1 or COM2).

When the RDP is being used as a Modbus serial slave device, you must also specify its Modbus device number in the project settings. See "Serial Slave Device Number," below.

After reading this section, proceed to "Modbus TCP Master Communication Settings" on page 38 if you will be using the RDP's Ethernet port to communicate with Modbus TCP slave devices. Otherwise, continue to the section titled "Configure Modbus I/O (PLC & RDP)" beginning on page 46.

NOTE: Before beginning this process, verify that RDP mode has been selected in the project's settings. See "Project Settings" beginning on page 15 for more information.

IMPORTANT: If you are only going to be using one of the drivers, set the unused driver's Enabled setting to 0. This turns off (disables) the driver. After making any changes to the RDP's serial or network communication settings, you must restart the RDP in order for the changes to take affect.

SERIAL SLAVE DEVICE NUMBER

When using the RDP as a Modbus serial slave device, you must specify the RDP's Modbus device number (1-254) in addition to configuring the communication parameters for the serial port being used for Modbus slave functionality (COM1 or COM2). This allows the RDP to operate as a serial slave device on RS-485 or radio networks where there is more than one slave device.

The device number is configured in the Settings dialog box.

- Select Settings from the File menu and enter the desired number in the Slave Device # field.

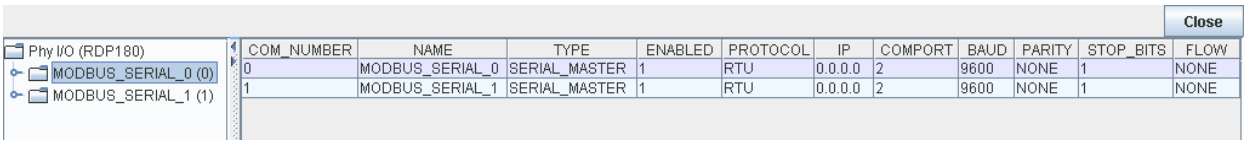
RDP: COM1

These settings are the ones the RDP uses when communicating with a Modbus master or slave device (for example, a RIO128) that is connected to the RDP's COM1 serial port. COM1 can be used in either RS-232 mode (Pins J1-2, J1-3, J1-5, J1-7, and J1-8) or RS-485 mode (Pins P2-4, P2-5, and P2-6).

IMPORTANT: You cannot connect devices to both the RS-232 pins *and* the RS-485 pins. COM1 is a single serial port that can be used in either RS-232 or RS-485 mode.

Review the serial specification of the device you are connecting to the RDP's serial port to determine the settings you should select.

Select the MODBUS_SERIAL_0 or MODBUS_SERIAL_1 driver from the tree. The parameters for the selected driver are listed in the first line of the table that appears in the right panel.



Note that the IP column is not applicable.

- **Com_Number** – The 0 (zero) in this field is the communications driver number. This field can't be edited.
- **Name** – The text in this field (MODBUS_SERIAL_0) is the name of this communications driver. This field can't be edited.

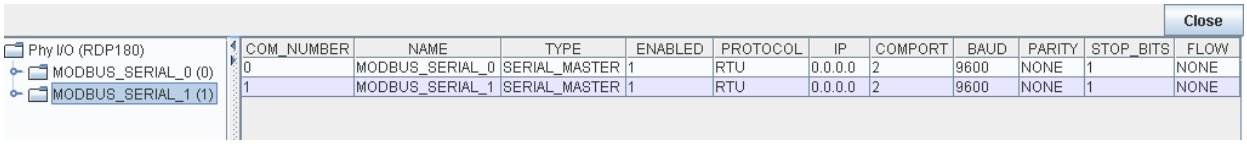
- **Type** – Select SERIAL_SLAVE or SERIAL_MASTER. When SERIAL_SLAVE is selected, the RDP will respond to queries from a Modbus master device over the COM1 port. When SERIAL_MASTER is selected, the RDP will poll Modbus slave devices over the COM1 port.
- **Enabled** – Select 1 from the drop-down list to indicate this port is being used.
- **Protocol** – Select RTU or ASCII from the drop-down list.
- **IP** – Not applicable.
- **Comport** – Select 1 from the drop-down list to indicate this driver is to be used for the COM1 port.
- **Baud** – Select 1200, 9600, 19600, or 38400 from the drop-down list. (Select 38400 for a RIO.)
- **Parity** – Select None, Odd, or Even from the drop-down list. (Select None for a RIO.)
- **Stop Bits** – Select 1, 2, or None from the drop-down list. (Select 1 for a RIO.)
- **Flow** – Select the method of hardware flow control for this device. Choose None, Radio (RTS-CTS), or Bus (CTS) from the drop-down list. (Select None for a RIO; select Radio for an RDR or other radio device; select Bus for an RS-485 device.)

RDP: COM2

These settings are the ones the RDP uses when communicating with a Modbus master or slave device that is connected to the RDP's COM2 serial port.

Review the serial specification of the device you are connecting to the RDP's serial port to determine the settings you should select.

Select MODBUS_SERIAL_0 or MODBUS_SERIAL_1 from the tree. The parameters for the selected driver are listed in the first line of the table that appears in the right panel.



Note that the IP column is not applicable.

- **Com_Numbe**r – The 1 (one) in this field is the communications driver number. This field can't be edited.
- **Name** – The text in this field (MODBUS_SERIAL_1) is the name of this communications driver. This field can't be edited.
- **Type** – Select SERIAL_SLAVE or SERIAL_MASTER. When SERIAL_SLAVE is selected, the RDP will respond to queries from a Modbus master device over the COM2 port. When SERIAL_MASTER is selected, the RDP will poll Modbus slave devices over the COM2 port.
- **Enabled** – Select 1 from the drop-down list to indicate this port is being used.
- **Protocol** – Select RTU or ASCII from the drop-down list.
- **IP** – Not applicable.
- **Comport** – Select 2 from the drop-down list to indicate this driver is to be used for the COM2 port.
- **Baud** – Select 1200, 9600, 19600, or 38400 from the drop-down list.
- **Parity** – Select None, Odd, or Even from the drop-down list.

- **Stop Bits** – Select 1, 2, or None from the drop-down list.
- **Flow** – Select the method of hardware flow control for this device. Choose None or Radio (RTS-CTS) from the drop-down list. (Select None for a RIO; select Radio for an RDR or other radio device.)

MODBUS TCP MASTER COMMUNICATION SETTINGS

Both the PLC033 and the RDP feature an Ethernet port that enables them to function as a Modbus TCP master and slave. As a Modbus TCP slave device, the PLC/RDP can respond to queries from a Modbus TCP master device. No configuration is required in PMT for this functionality.

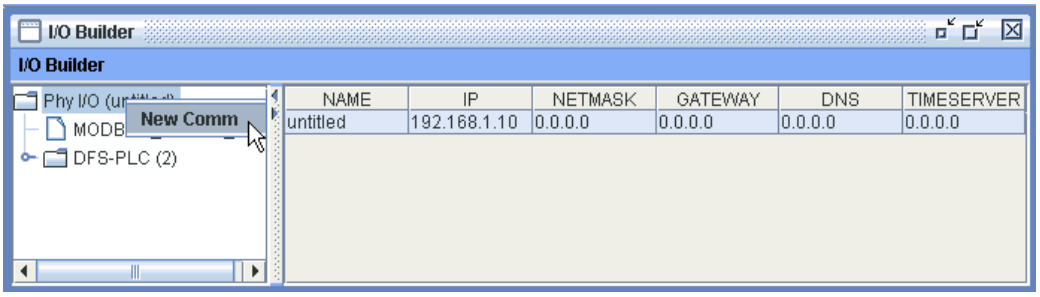
As a Modbus TCP master device, the PLC/RDP can communicate with up to four Modbus slave devices. Each device must have a unique IP address. To enable the PLC/RDP to function as a Modbus TCP master and communicate with slave devices, a Modbus TCP communications driver must be added and configured in PMT for each device. For example, to communicate with two slave devices, you would add two drivers and configure each with the appropriate IP address.

After reading this section, proceed to "Configure Modbus I/O (PLC & RDP)" beginning on page 46.

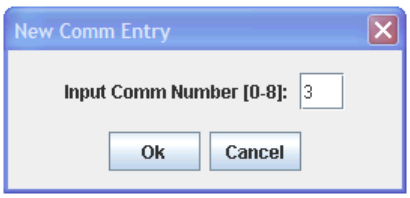
Add and Configure a Modbus TCP Driver

Note: You must add and configure a driver for *each* Modbus TCP slave device that the PLC/RDP will be communicating with (up to four devices). Each device must have a unique IP address.

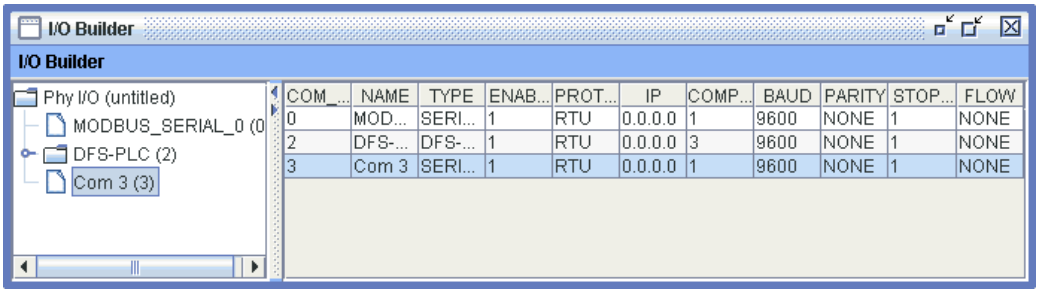
1. Open I/O Builder.
2. Right click Phy I/O.
3. Click **New Comm**.



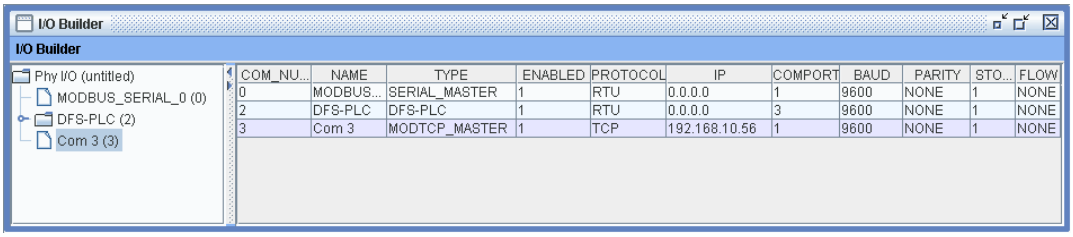
4. The communications driver numbers are automatically incremented by 1 (one) each time a driver is added. Therefore, the next driver added will be added at Com 4, the next at Com 5, the next at Com 6. If desired, you can enter a different number (any number between 3 and 8) in the **New Comm Entry** dialog box



5. Click **OK**. A new driver is added.



6. For each Modbus TCP driver, enter values for **Name** (optional), **Type**, **Enabled**, **Protocol**, and **IP**. The **Comport**, **Baud**, **Parity**, **Stop Bits**, and **Flow** fields are not applicable.



- **Com_Number** – The number (3, 4, 5, or 6) in this field is the communications driver number. This field can't be edited.
- **Name** – The text in this field (MODBUS_SERIAL_1) is the name of this communications driver. This field can be edited or left at its default value.
- **Type** – Select MODTCP_MASTER. When MODTCP_MASTER is selected, the PLC/RDP will be able to query a Modbus slave device over the network.
- **Enabled** – Select 1 from the drop-down list to indicate this communications driver is being used.
- **Protocol** – Select TCP from the drop-down list.
- **IP** – The IP address of the Modbus TCP slave device that the PLC/RDP will be querying.

After reading this section, proceed to "Configure Modbus I/O (PLC & RDP)" beginning on page 46 for instructions on adding devices and I/O to Modbus drivers.

IMPORTANT: After making any changes to the PLC/RDP's serial or network communication settings, you must restart the PLC/RDP in order for the changes to take affect.

CONFIGURE DFS I/O (PLC033 ONLY)

Each slot in the RTU's modular backplane has a letter designation (A, B, C, etc.). Before starting this process, either examine the RTU or review schematics, and note the letter of the slot that each module is installed in (e.g., slot A = AMM002, slot B = DMM002). When you add modules in I/O Builder, you must add them in the order in which they are installed in the RTU. The letter designation in I/O Builder *must* match what is found in the RTU.

If the PLC Central mode function is to be used, all remote I/O that is to be polled during PLC Central mode must be configured under the REMOTE branch of the DFS_PLC (2) node. The procedure for adding and configuring remote I/O for PLC Central functionality is similar to the procedure provided here with a few minor differences. Review the information provided in the ***PLC033 Installation and Operation Manual*** for more details.

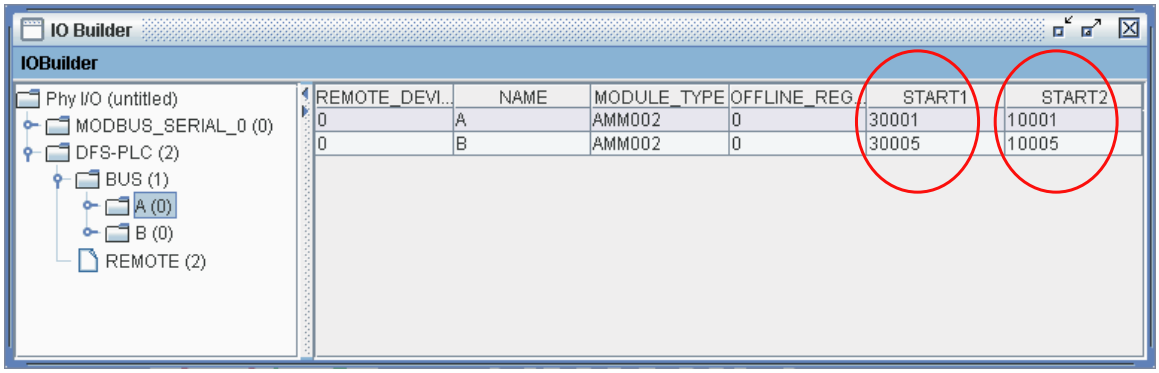
REGISTER MAPPING

The PLC at its core is a Modbus device. Modbus is an industry-standard protocol and as such enables the PLC to communicate with any third-party Modbus devices, including Open Control Solutions' RIO128 and RIO032 rail-mounted I/O devices. The PLC is able to support both DFS and Modbus protocols through a process called mapping. In mapping, each physical I/O point is assigned to a unique register in the PLC's Logical Memory Map. The Logical Memory Map is comprised of four register ranges:

I/O Type	Register Range
Digital Outputs (Coils)	00001-09999
Digital Inputs (Discrete Inputs)	10001-19999
Analog Inputs (Input Registers)	30001-39999
Analog Outputs (Holding Registers)	40001-49999

Before configurations can be installed in the PLC, the physical I/O configured in I/O Builder must be mapped to the correct register range in the PLC's Logical Memory Map. By mapping all points, both DFS-type points and Modbus-type points, to a common map using a common format, you are able to easily create logic for devices that use different communication protocols. When the PLC receives a message for a particular (logical) register, it searches the Logical Memory Map for the physical I/O point that is mapped to the logical register.

DFS-type points are mapped to the Logical Memory Map by assigning each module a start register. Some modules require two start registers, because they have two types of I/O. For example, the AMM002 has four analog inputs and four digital inputs. Each point type must be mapped to the correct register range. START1 would be the starting point for the first analog input, and the remaining points would be placed in consecutive registers after the register specified in START1. START2 would be the starting point for the first digital input, and the remaining digital inputs would be placed in consecutive registers after the register specified in START2. For modules that have only one type of I/O, you only have to specify a START1 register; a -1 will appear in the START2 field.



A different method is used for mapping the points of devices that use Modbus protocol. That method is discussed in the section titled "Configure Modbus I/O" beginning on page 46.

Additionally, if you will be polling the PLC from an HT3 system (for example, a Hyper SCADA Server), or if you will be using the PLC033 as a PLC Central, you will need to map the registers into DFS module points. This procedure is discussed in the section titled "DFS Radio Mapping" beginning on page 56.

IMPORTANT: You cannot map digital outputs to the 9900-9999 range or analog outputs to the 49900-49999 range. Those registers are reserved for the PLC's Special Function Registers. A listing of the SFRs can be found on page 5 in Chapter 1: Product Overview.

PLC033 Set Point Variables (Q Points)

When a PLC project is created, 168 Q points that represent the PLC033's 168 unused memory locations are automatically added to the Logical Memory Map. Each point is given a label that begins with the letter Q (e.g., Q1, Q2, Q3, etc.). The Q point registers reside in the 49000 range beginning at register 49464 and ending at register 49798. Because Q points are 32-bit floating point values, each point requires two registers, and each Q point begins on an even register (e.g., Q point number 50 resides at registers 49562 and 49563).

To add Q points to an existing PLC project, you must change the project type to RDP and then change it back to PLC. If you have I/O in the Logical Memory Map that is mapped to the registers assigned to Q points, they will not be overwritten. The Q points will be placed "around" the taken registers and will always start at an even register number.

For more information on Q points, see "Q Points (PLC033 Only)" on page 6. Information can also be found in the ***PLC033 Installation and Operation Manual*** and in the online documentation for HT3.

Offline Register

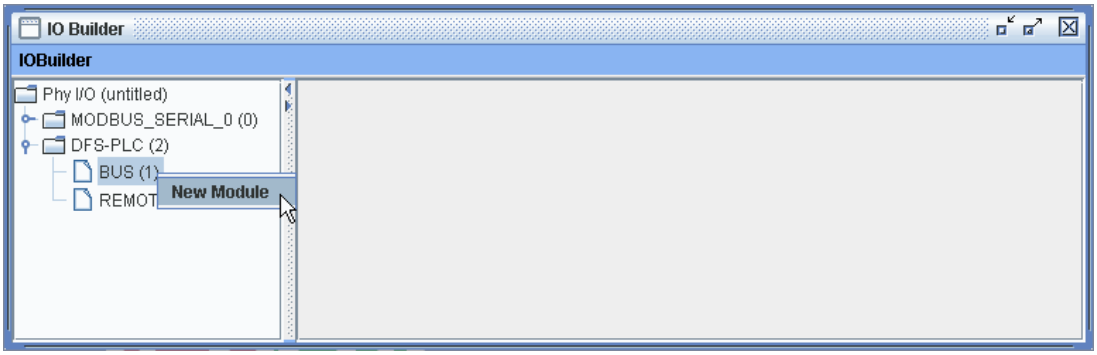
The PLC033 does not require that physical modules be mapped to equivalent module addresses in the DFS Radio Map. For efficiency purposes you may choose to reduce the number of modules and I/O responding over the radio link by mapping only desired I/O to Radio Modules. An optional Offline Register field is provided so that a digital register can be used to indicate that a physical module has gone offline.

The Offline Register must be in the writeable range (00001-09999; it is recommended that a contiguous bank of 15 registers be reserved for this purpose). The offline register will be FALSE (0) if the associated hardware module is communicating properly on the bus. Otherwise it is TRUE (1). When the Offline Register is TRUE the PLC033 will ***not*** respond to radio messages for the associated module causing it to go offline in telemetry.

If the Offline Register field is set to 0 (zero) or left blank, there is ***no*** offline handling of the hardware modules and the PLC033 will respond with the last known values of the associated module.

ADD A DFS MODULE

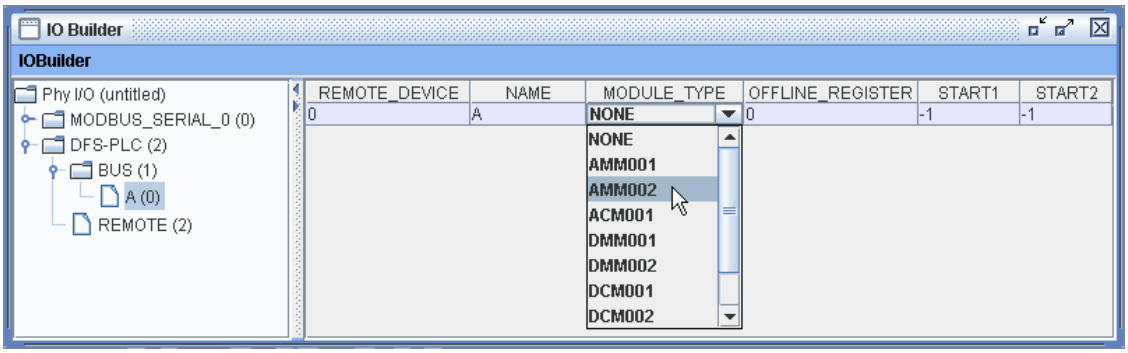
1. Expand the tree, so that BUS (1) is visible. Select BUS (1) and right click.
2. Select **New Module** from the pop-up menu. Module A is added to the tree, and the Module Table is displayed in the right panel. The next module added will be given the letter B, the next C, and so on.



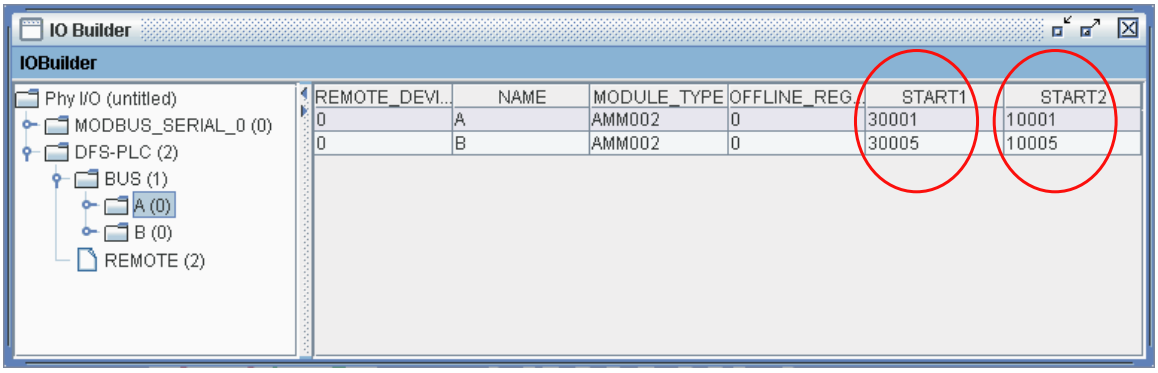
3. In the **New Module Entry** dialog box, type the letter of the module being added and click OK.

The module is added to the tree; the module's properties are displayed in the right panel. (Note that as modules are added, the letter in the New Module Entry dialog box will automatically increment by one letter.)

4. In the right panel, select the correct module type from the drop-down list (see Table 3-1: DFS Function Modules, next page, for a list of modules and their corresponding I/O types).



5. You can leave the modules at the default registers assigned by I/O Builder, or you can map them to different registers by typing register numbers in **START1** (and **START2** if necessary). If you enter a register that has already been used or is assigned to a Special Function Register (see note below), a warning box asking you to modify the starting points will appear. (Review the information in "Register Mapping" on page 40.)



6. Enter the register that will monitor the module's status in the **Offline Register** field. The register selected must be in the 00001-09999 range. If you enter a register that has already been used or is assigned to a Special Function Register (see note below), a warning box asking you to select another register will appear. If you don't want to use this feature, leave the default value of 0 in the **Offline Register** field. (Review "Offline Register" on page 41.)

IMPORTANT: When selecting starting registers to map physical I/O to the PLC033's Logical Memory Map, be aware that certain registers have been set aside for special functions (for example, Global Alarm and Maximum Ladder Loop Time). A list of these registers can be found in Chapter 1: Product Overview.

Table 3-1: DFS Function Modules

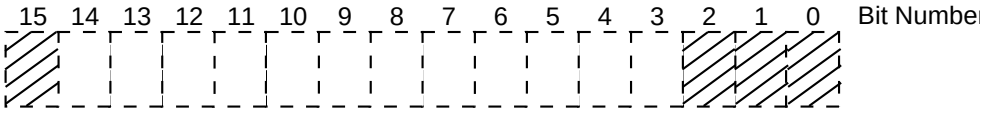
Module Name	No. of Analog Inputs	No. of Analog Outputs	No. of Digital Inputs	No. of Digital Outputs	Configure As
AMM001*	4		4		AMM001
AMM002	4		4		AMM002
ACM001*		4			ACM001
ACM002		4			ACM001
DMM001*			12		DMM001
DMM002			12		DMM002
DCMOO1*			4	8	DCM001
DCM011*			4	8	DCM001
DCM012*			8	4	DCM002
DCM002*			8	4	DCM002
DCM003-1			4	8	DCM003-1

Module Name	No. of Analog Inputs	No. of Analog Outputs	No. of Digital Inputs	No. of Digital Outputs	Configure As
DCM003-2			8	4	DCM003-2
DCM003-3			4	8	DCM003-1
DCM003-4			8	4	DCM003-2
DCM003-5			4	8	DCM003-1
DCM003-6			8	4	DCM003-2

* Legacy modules

CONFIGURE DFS MODULE I/O

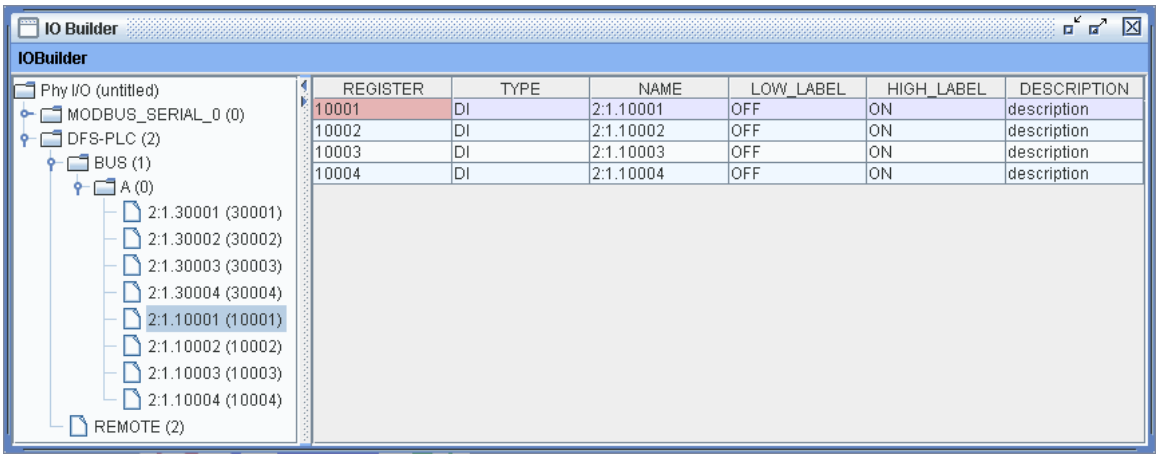
IMPORTANT: All DFS analog modules are 12 bit. These 12 bits are placed in bits 3-14 of the I/O register as illustrated below.



1. Expand the tree so that the modules below BUS (1) are visible.
2. Double click the module you want to configure. The tree expands so that all available points for that module type are visible.
3. For each point, enter the values for the parameters corresponding to the point type – digital (see next page) or analog (see page 45):

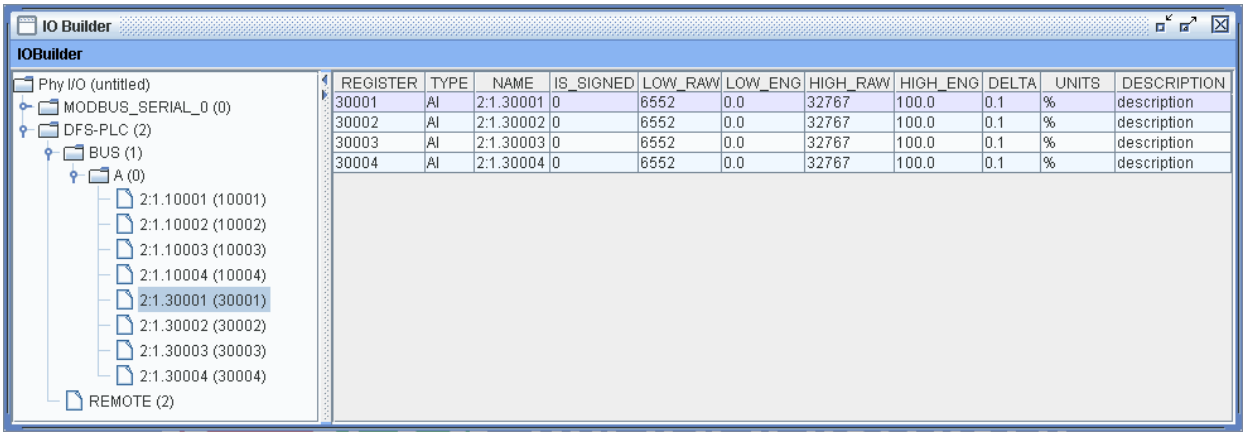
IMPORTANT: Each register should be given a unique and meaningful name, referred to as a tag name. These tag names can be helpful when creating ladder logic programs and building custom screens, since you can select a register based on its tag name rather than its number.

DIGITAL POINTS



- **Register** – Leave at its current value.
- **Type** – Uneditable label describing the type of point being configured (DI: digital input; DO: digital output).
- **Name** – Give the point a meaningful name. Up to 20 characters can be entered in this field. Valid characters are the letters A-Z, the numbers 0-9, and the characters . (period), / (forward slash), _ (underscore), @ (at symbol), and $ (dollar sign).
- **Low Label** – Type a label that describes the point's "low" state.
- **High Label** – Type a label that describes the point's "high" state.
- **Description** – Brief description of the point. Up to 40 characters can be entered in this field. Valid characters are the letters A-Z, the numbers 0-9, and the characters . (period), / (forward slash), _ (underscore), @ (at symbol), and $ (dollar sign).

ANALOG POINTS



- **Register** – Leave at its current value.
- **Type** – Uneditable label describing the type of point being configured (AI: analog input; AO: analog output).

- **Name** – Give the point a meaningful name. Up to 20 characters can be entered in this field. Valid characters are the letters A-Z, the numbers 0-9, and the characters . (period), / (forward slash), _ (underscore), @ (at symbol), and $ (dollar sign).
- **Is Signed** – Flag that allows negative raw and engineering values. 1 = True (Yes); 0 = False (No).
- **Low Raw** – Minimum expected raw value for the point; the value as directly read from the device without any scaling, or conversion. Enter a value between 0 and 32767*.
- **Low Eng** – Minimum desired scaled (or converted) engineering value for the input. This value corresponds to the Low Raw value defined above and is used for display and reporting purposes.
- **High Raw** – Maximum expected raw value for the point; the value as directly read from the device without any scaling, or conversion. Enter a value between 0 and 32767*.
- **High Eng** – Maximum desired scaled (or converted) engineering value for the input. This value corresponds to the High Raw value defined above and is used for display and reporting purposes.
- **Delta** – Minimum change to be reported (in engineering units). A change in the value of this point will only be reported if the change is equal to or greater than Delta.
- **Units** – Units of measurement for this point (e.g., feet or gallons).
- **Description** – Brief description of the point. Up to 40 characters can be entered in this field. Valid characters are the letters A-Z, the numbers 0-9, and the characters . (period), / (forward slash), _ (underscore), @ (at symbol), and $ (dollar sign).

4. When you are finished configuring DFS module I/O, you can transfer the configurations to the PLC by choosing **Install** from the File menu. Note that this only installs the I/O configurations. Ladders and screens are installed using a separate process.

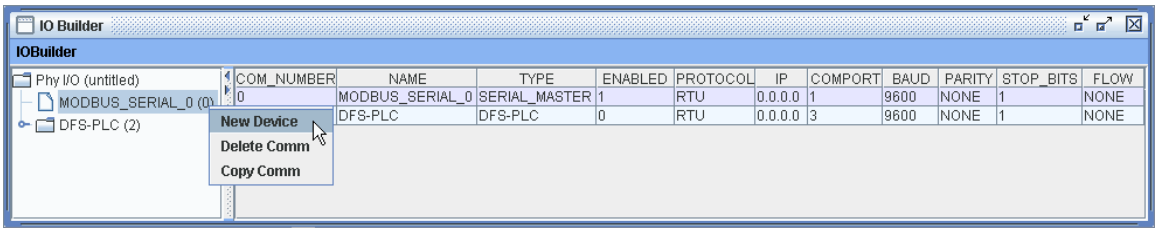
   Observe the PLC033's status LED. As it processes the configurations, the blinking pattern will be 3 quick flashes followed by a pause. The status LED will then turn off for a few seconds. When the installation is complete, the status LED will return to its normal blinking pattern (slowly turns on and then off).

CONFIGURE MODBUS I/O (PLC & RDP)

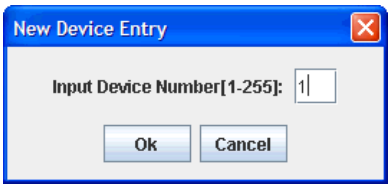
One RS-232 master or slave device can be added to the PLC or RDP's COM1 port. There is no limit on slave devices when using the COM1 port in RS-485 mode. Several RS-485 devices can be daisy chained together, but be aware that each device added will affect the polling rate. The RDP's COM2 port can interface with one RS-232 master or slave device. Usually, this device is a radio through which the RDP polls remote I/O. However, it is also possible to use the COM2 serial port as an interface to a single RS-232 master or slave device.

ADD A NEW MODBUS DEVICE

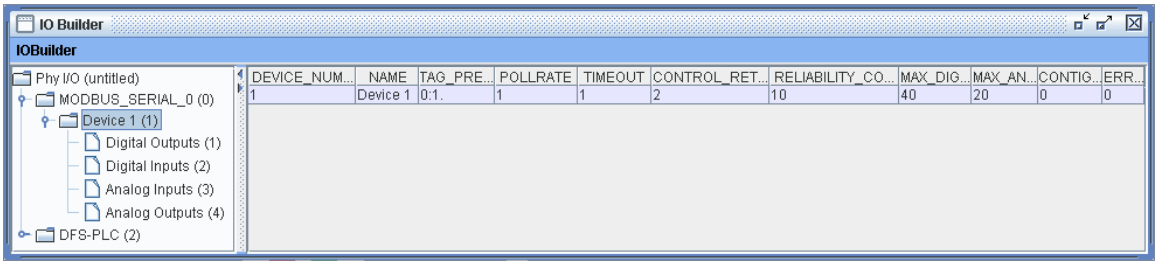
1. Select the desired Modbus driver on the tree. Note that MODBUS_SERIAL_0 is used for a PLC033 or RDP; MODBUS_SERIAL_1 is only used for an RDP.
2. Right click and select **New Device** from the pop-up menu.



3. Enter the number for the device in the **New Device Entry** box.



The device will be added to the tree below the selected device, and the Device Table will be displayed in the right panel.



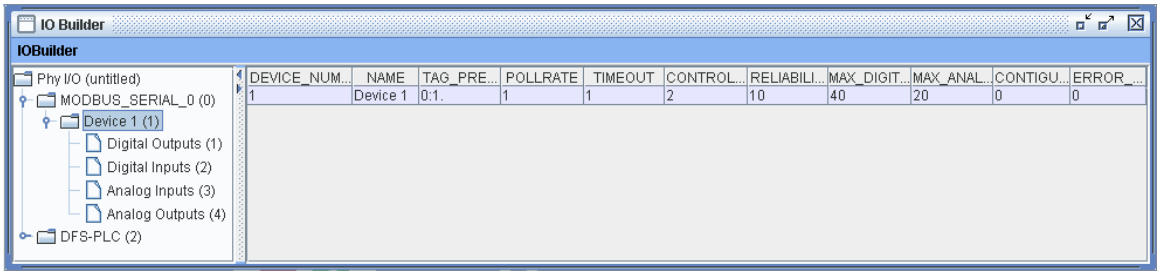
DEVICE NAME AND TAG PREFIX

Each device added to the PLC033's COM1 port or the RDP's COM1 and COM2 ports can be given a name (for example, RIO) and a tag prefix (for example, STN1 or MAIN) that uniquely identifies it. The tag prefix and device name will appear before the name of each point configured under the device and helps anyone looking at the configuration to easily identify what device the configuration is for. It can also help someone developing custom screens and ladder logic programs to identify the device. As an example, points with a device name of RIO and a tag name of Stn1 would appear as Stn1_RIO_X (where X represents the register number). If a tag name is not assigned, each point will have a prefix of A:B., where A represents the driver number and B represents the device number. For example, 0:1. for the first device configured under the MODBUS_SERIAL_0 (0) driver.

Instructions for providing a name and tag prefix are included in the next section, "Configure a Modbus Device."

CONFIGURE A MODBUS DEVICE

Select the device to be configured and enter values for the following parameters:



- **Device Number** – Leave at its current value unless it needs to be changed.
- **Name** – A meaningful name for this device. Up to 20 characters can be entered in this field. Valid characters are the letters A-Z, the numbers 0-9, and the characters . (period), / (forward slash), _ (underscore), @ (at symbol), and $ (dollar sign). In the tree on the left, the name of the device changes to reflect the name provided here.
- **Tag Prefix** – A prefix that uniquely identifies this device. When points are added to this device, the name of each point will begin with the prefix entered here. This facilitates selecting points when creating ladder logic, custom screens, and historical trends, because you can locate a point based on its tag instead of searching for an arbitrary number. Up to eight characters can be entered in this field. Valid characters are the letters A-Z, the numbers 0-9, and the characters . (period), / (forward slash), _ (underscore), @ (at symbol), and $ (dollar sign).
- **Poll Rate** – How often (in seconds) this device should be polled (e.g., every 5 seconds).
- **Time Out** – Length of time (in seconds) to wait for a device to reply before reporting an error. Valid range is 1-15,000 seconds.
- **Control Retry** – Number of control attempts allowed before a control is considered failed. Select a value between 0 and 9 from the drop-down list.
- **Reliability Count** – Number of consecutive polling errors allowed before an offline status is returned. Select a value between 0 and 10 from the drop-down list.
- **Max Digital** – Maximum number of digital points to be polled in a single message.
- **Max Analog** – Maximum number of analog points to be polled in a single message.
- **Contiguous Only** – Flag that indicates that this device's points must be polled in blocks of contiguous addresses (Contiguous Only = 1); Or, that non-contiguous polling is allowed (Contiguous Only = 0). For example, if you have a RIO032 connected to the PLC and are only using digital points 1, 2, 3, 4, and 10, and you have Max Digital set to 5, and Contiguous Only set to 1 (true), the PLC will poll points 1, 2, 3, and 4 in one poll and then will poll point 10 separately. That is, it can only poll up to 5 points if they are consecutively numbered. If Contiguous Only was set to 0 (false), then all five points would have been polled in one message. **IMPORTANT**: Only set this flag to false (0), if the device being polled allows the "gaps" between points to be polled. Some devices will issue an error when an attempt is made to poll non-contiguous points.
- **Error Log** – Future use.

ADD AND CONFIGURE MODBUS I/O

IMPORTANT: Each point should be given a unique name, referred to as a tag name. These tag names make it easier to locate (or reference) the I/O when creating ladder logic programs and building custom screens.

RIO WIZARDS

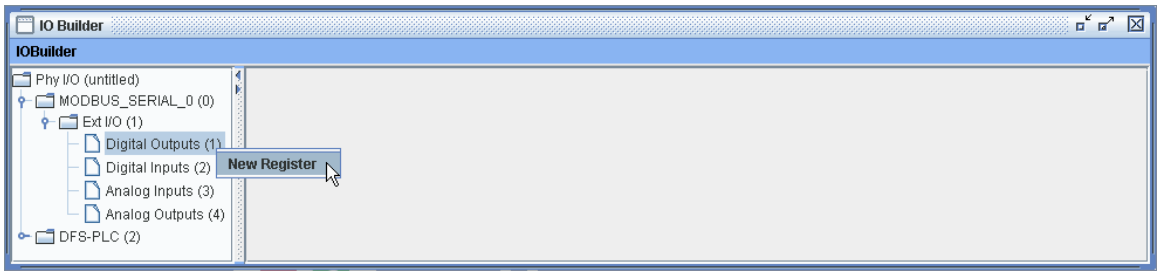
If the device you are adding is a RIO032 or RIO128, you can use the **RIO 32 Wizard** or the **RIO 128 Wizard** to add all of the appropriate I/O to this device at once instead of adding each point individually.

After adding and configuring the device, right click the device name in the tree and select the correct wizard from the pop-up menu. When you are finished, you can configure the individual I/O as discussed in the next two sections ("Digital Inputs and Outputs" and "Analog Inputs and Outputs").

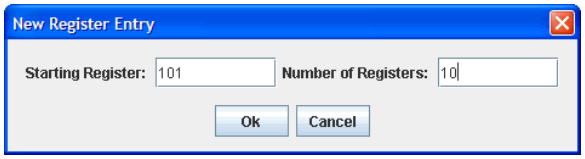
DIGITAL INPUTS AND OUTPUTS

NOTE: Skip steps 1-3 if you used one of the RIO wizards.

1. Select the Digital Outputs or Digital Inputs node under the device that you want to add the inputs/outputs to.
2. Right click and select **New Register** from the pop-up menu.

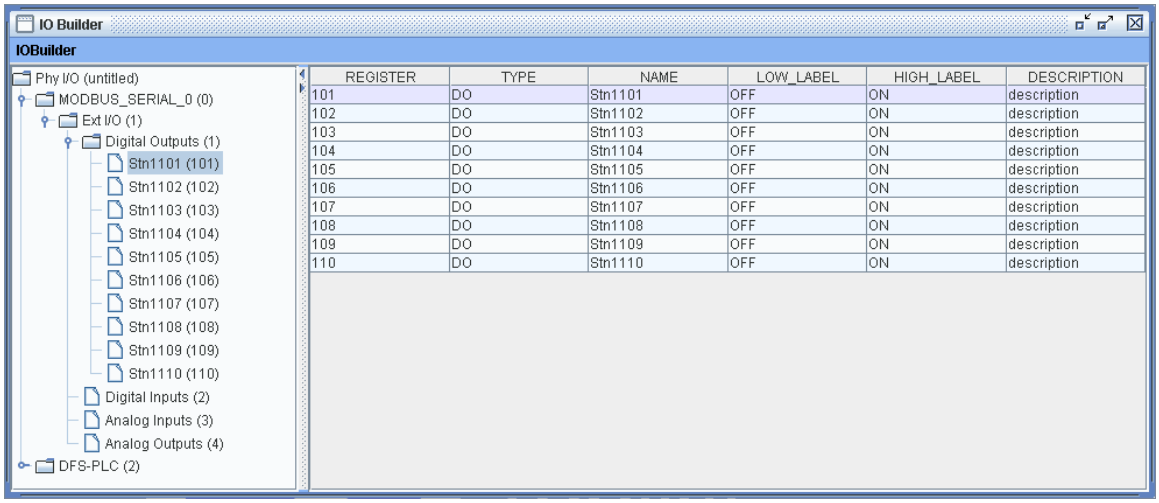


3. In the **New Register Entry** box, enter values for **Starting Register** and **Number of Registers** and click **OK**.



Starting Register is the number assigned to the first register. The number of each subsequent register will be increased by 1 (one) until the total number of registers is reached. The registers will be added to the tree below the selected device. The parameters of the registers will be displayed in the right panel.

4. For each register, enter values for the following parameters:

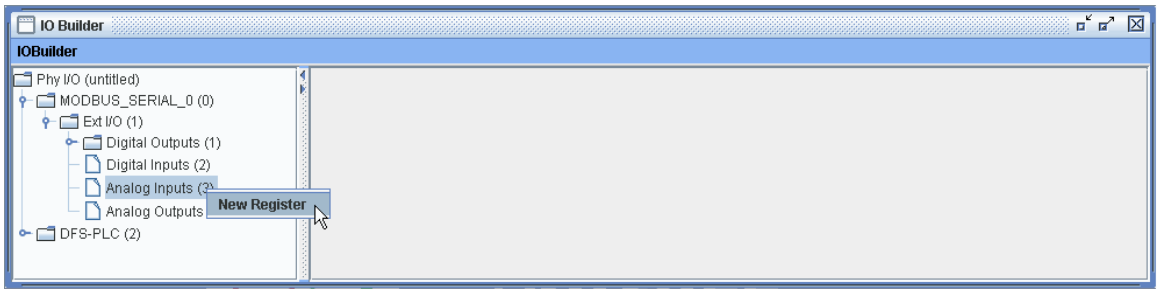


- **Register** – Leave at its current value unless it needs to be changed.
- **Type** – Uneditable label describing the type of point being configured (DI: digital input; DO: digital output).
- **Name** – Give the point a meaningful name. Up to 20 characters can be entered in this field. Valid characters are the letters A-Z, the numbers 0-9, and the characters . (period), / (forward slash), _ (underscore), @ (at symbol), and $ (dollar sign).
- **Low Label** – Label that describes the point's "low" state. Select a label from the drop-down list or type one of your own.
- **High Label** – Label that describes the point's "high" state. Select a label from the drop-down list or type one of your own.
- **Description** – Brief description of the point. Up to 40 characters can be entered in this field. Valid characters are the letters A-Z, the numbers 0-9, and the characters . (period), / (forward slash), _ (underscore), @ (at symbol), and $ (dollar sign).

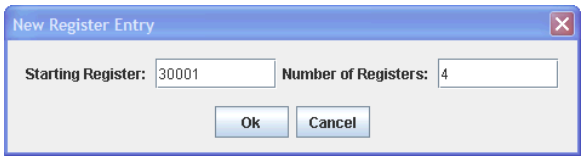
ANALOG INPUTS AND OUTPUTS

NOTE: Skip steps 1-4 if you used one of the RIO wizards.

1. Select the Analog Inputs or Analog Outputs node under the device that you want to add the inputs/outputs to.
2. Right click and select **New Register** from the pop-up menu.

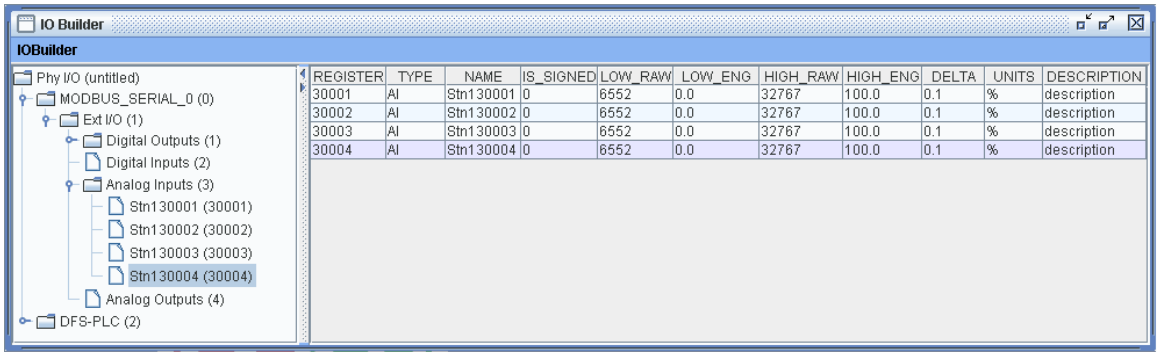


3. In the **New Register Entry** box, enter values for **Starting Register** and **Number of Registers** and click **OK**.



Starting Register is the number assigned to the first register. The number of each subsequent register will be increased by 1 (one). The registers will be added to the tree below the selected device. The parameters of the registers will be displayed in the right panel.

4. For each register, enter values for the following parameters:



- **Register** – Leave at its current value unless it needs to be changed.
- **Type** – Uneditable label describing the type of point being configured (AI: analog input; AO: analog output).
- **Name** – Give the point a meaningful name. Up to 20 characters can be entered in this field. Valid characters are the letters A-Z, the numbers 0-9, and the characters . (period), / (forward slash), _ (underscore), @ (at symbol), and $ (dollar sign).
- **Is Signed** – Flag that allows negative raw and engineering values. 1 = True (Yes); 0 = False (No).
- **Low Raw** – Minimum expected raw value for the point; the value as directly read from the device without any scaling, or conversion. Enter a value between 0 and 65520.*
- **Low Eng** – Minimum desired scaled (or converted) engineering value for the input. This value corresponds to the Low Raw value defined above and is used for display and reporting purposes.
- **High Raw** – Maximum expected raw value for the point; the value as directly read from the device without any scaling, or conversion. Enter a value between 0 and 65520.*
- **High Eng** – Maximum desired scaled (or converted) engineering value for the input. This value corresponds to the High Raw value defined above and is used for display and reporting purposes.
- **Delta** – Minimum change to be reported (in engineering units). A change in the value of this point will only be reported if the change is equal to or greater than Delta.
- **Units** – Units of measurement for this point (e.g., feet or gallons).
- **Description** – Brief description of the point. Up to 40 characters can be entered in this field. Valid characters are the letters A-Z, the numbers 0-9, and the characters . (period), / (forward slash), _ (underscore), @ (at symbol), and $ (dollar sign).

* If the data from this register will be transmitted using a DFS protocol (RIM protocol – radio link; NIM RTU protocol – network interface), you may need to specify a bit shift other than the default in order to accommodate the maximum raw value you expect this register to have. See "DFS Radio Mapping: Register (Bit) Shifting Options" on page 57 for more information.

MAPPING MODBUS I/O TO THE LOGICAL MEMORY MAP

The PLC at its core is a Modbus device. Modbus is an industry-standard protocol and as such enables the PLC to communicate with any third-party Modbus devices, including Open Control Solutions' RIO128 and RIO032 rail-mounted I/O devices. In mapping, each physical I/O point is assigned to a unique register in the PLC's Logical Memory Map. The Logical Memory Map is comprised of four register ranges:

I/O Type	Register Range
Digital Outputs (Coils)	00001-09999
Digital Inputs (Discrete Inputs)	10001-19999
Analog Inputs (Input Registers)	30001-39999
Analog Outputs (Holding Registers)	40001-49999

Before configurations can be installed in the PLC, the physical I/O configured in I/O Builder must be mapped to the correct register range in the PLC's Logical Memory Map. When the PLC receives a message for a particular (logical) register, it searches the Logical Memory Map for the physical I/O point that is mapped to the logical register.

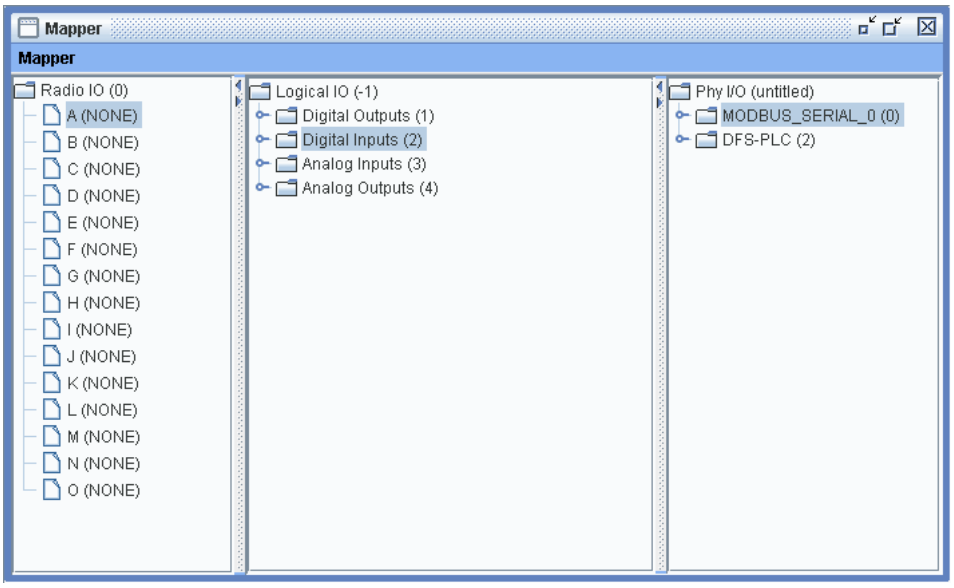
Modbus-type registers are mapped to the Logical Memory Map using Mapper.

IMPORTANT: You cannot map digital outputs to the 9900-9999 range or analog outputs to the 49900-49999 range. Those registers are reserved for the PLC's Special Function Registers. A listing of the SFRs can be found on page 3 in Chapter 1: Product Overview.

A different method is used for mapping the points of DFS function modules. That method is discussed in the section titled "Configure DFS I/O (PLC033 Only): Register Mapping" beginning on page 40.

LAUNCH MAPPER

1. Select **Mapper** from the **Build** menu. The mapping tool opens in a new window.
2. This utility consists of three panels.



The left panel represents the DFS Radio Map. It is used for mapping registers that will be polled from an HT3 server using a DFS protocol (RIM protocol – radio link; NIM RTU protocol – network interface). See "DFS Radio Mapping" on page 56 for more information.

The middle panel represents the Logical Memory Map. It has four nodes representing the four types of I/O (digital outputs, digital inputs, analog inputs, and analog outputs). If DFS modules have already been configured (PLC033 only), they will be listed below their corresponding node (you will need to expand the node to view them). Additionally, the PLC's Special Function Registers will appear under the Digital Output node and the Analog Output node. These registers are colored orange, so they can be easily identified.

The right panel represents the physical I/O that has been configured. This includes all DFS function modules configured for the DFS PLC driver (PLC033 only) and devices configured for the Modbus Serial driver(s). Although the DFS PLC driver is listed, it can't be expanded. This is because those points have already been mapped (when they were initially configured).

When you are finished configuring and mapping Modbus I/O, install the configurations on the PLC using the procedure on page 61.

MAPPING METHODS

There are two ways to accomplish mapping: map an entire device or map each point individually.

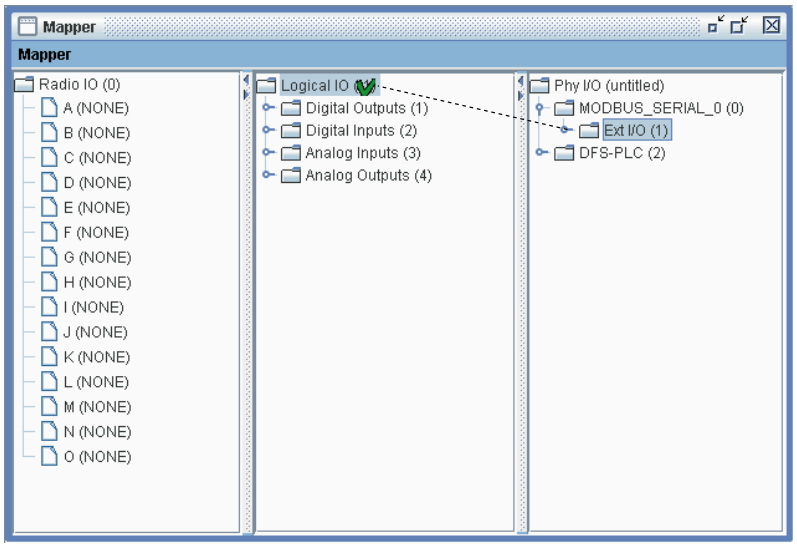
MAP AN ENTIRE DEVICE

When mapping an entire device, you can use 1-1 Mapping or Custom Mapping.

- **1-1 Mapping** – In 1-1 Mapping, each of the device's registers is mapped to the "logical" register that matches its "physical" register. For example, if the device's first digital output resides in register 1 in the physical device, it will also reside in register 1 in the Logical Memory Map.

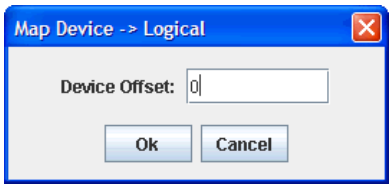
- **Custom Mapping** – If you can't do 1-1 Mapping because other physical registers have already been mapped to those logical registers, you can use custom mapping and specify an offset. For example, if you were to enter 500 for the offset, the logical register would be equal to its "physical" register plus 500. If the first digital output resided in register 1 on the physical device, it would reside in register 501 in the Logical Memory Map.

1. On the Physical I/O tree (right panel), expand the Modbus Serial driver node to view its devices.
2. Select the device you want to map. Hold down the mouse button and drag the device to the top node of the Logical Map tree (middle panel).



A green check mark will appear to indicate it is OK to map to this location. If you try to map a device onto a node (e.g., the digital outputs node), a red X will appear.

3. In the **Map Device -> Logical** dialog box, select the type of mapping you want to do by entering an offset.

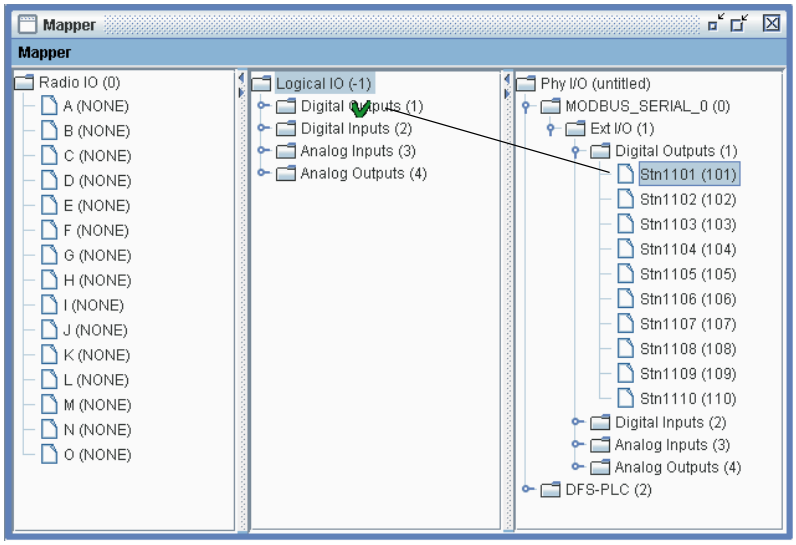


For 1-1 mapping, enter an offset of 0; for custom mapping, enter the desired offset. Click **OK**.

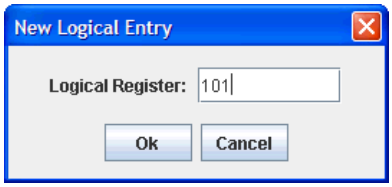
4. When you are finished configuring and mapping Modbus I/O, install the configurations on the PLC using the procedure on page 61.

MAP AN INDIVIDUAL REGISTER

1. On the Physical I/O tree (right panel), expand the Modbus Serial driver node to view its devices. Expand the node of the device you want to map and then expand the node of the I/O type you want to map so that all of the registers are visible.
2. On the Physical I/O tree, click the first point you want to map. Hold the mouse button down and drag the cursor over the name of the correct I/O type on the Logical Memory Map tree. For example, if you are mapping a digital output, drag the cursor over to the Digital Outputs node.



3. In the **New Logical Entry** dialog box, enter the logical register number that you want to map this physical register to. Click **OK**.



4. When you are finished configuring and mapping Modbus I/O, install the configurations on the PLC using the procedure on page 61.

DFS RADIO MAPPING

Before you can begin polling the PLC (and all devices connected to it) from an HT3 server using a DFS protocol (RIM protocol – radio link; NIM RTU protocol – network interface), you must map the desired registers into DFS modules [maximum 30 modules; up to 15 modules per station with a two (2) station maximum]. Note that it isn't necessary to map *all* of the registers – only those that will be sent to the HT3 central using a DFS protocol.

The PLC033 can emulate up to two DFS RTUs. In addition to the default Radio I/O branch in the Radio Map, a second "remote" device can be added. This remote device can also accommodate up to 15 DFS modules. Adding a "remote" device allows developers to exceed the 15 module maximum for DFS RTUs.

Radio mapping is accomplished through the Mapper. Refer to "Launch Mapper" on page 52 for an overview of this tool.

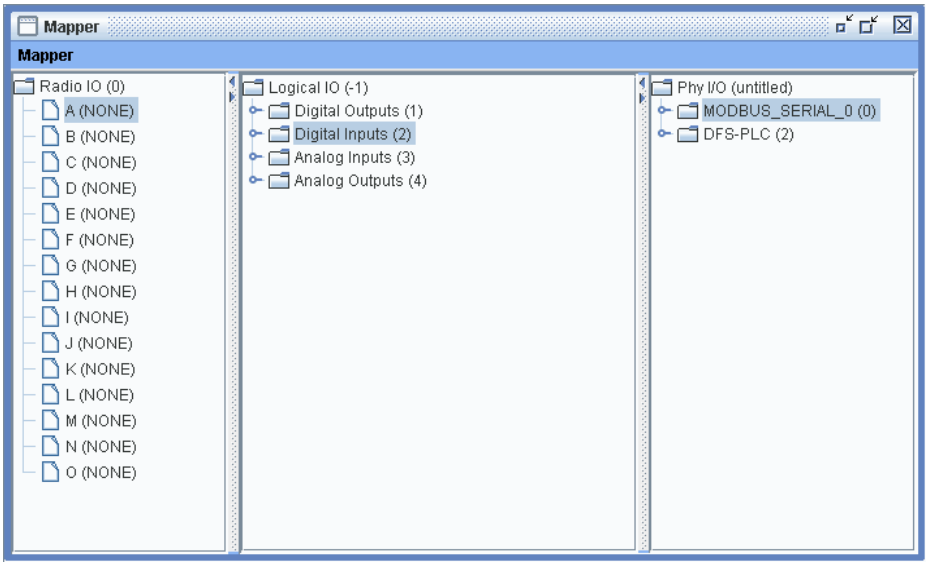
ADDING A REMOTE DEVICE (EMULATED DFS RTU)

A remote device (emulated DFS RTU) can be added to the Radio Map by doing the following:

1. Right-clicking the main (Radio Map) branch.
2. Select Add Remote Device.
3. When prompted, enter a Remote Device Number. The Remote Device Number should be the next station number after the physical address of the station the PLC is installed in (for example, if the PLC is in RTU #17 then add remote device #18 in the Radio Map.
4. Configure the remote device (emulated station) in HT3. The emulated station's RIM type should be configured as a RIM004 in HT3. The station number in HT3 must match that configured for the remote device in PMT.

In our example, the RTU would now respond as station 17 *and* 18 in HT3.

RADIO I/O TREE



The Radio I/O tree, which is a list of 15 DFS modules, appears in the left panel of Mapper. A second set of 15 modules can be added to the Radio I/O tree by right-clicking Radio I/O and selecting Add Remote Device.

To map the registers into DFS modules, you must add modules that can accommodate all of the desired registers. You must add modules of the correct type and in sufficient numbers to accommodate the registers. A good first step is to determine how many of each I/O type you have (i.e., digital input, digital output, analog input, analog output), then figure out how many DMMs, AMMs, DCMs, etc., that you need. Refer to the table on page 43 to see how many points and of what type each module has.

REGISTER (BIT) SHIFTING OPTIONS

The DFS RIM and NIM RTU protocols are 12-bit protocols with a full-scale output of 4095. You can specify how you want to shift the bits in the register when working with 16-bit data from Modbus-compatible devices or from logical I/O created in ladder logic that will be sent over the DFS radio link using RIM protocol or via the network using NIM RTU protocol.

- More shift allows larger numbers, but it provides less accuracy.
- Less shift provides more accuracy, but it gives a smaller full-scale output.

Consider the tradeoffs associated with each option before selecting a bit shift setting. Choose the option that makes the most sense in your situation.

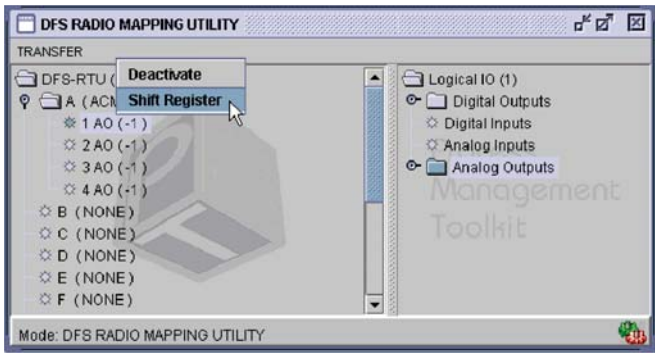
DFS radio mapping provides 5 register shift settings. Accuracy is lost as you move from 0 to 4 bit shift, but the full-scale output increases.

- 0 Bit Shift – This is the default setting. This setting provides the greatest accuracy, because it counts by 1's. However, you are limited to a full-scale output of 4095. This setting would be appropriate for a counter, because accuracy – not a large number – is the most important element.
- 1 Bit Shift – Slightly less accurate than 0 Bit Shift, this setting counts by 2's and provides a full-scale output of 8190.
- 2 Bit Shift – This setting counts by 4's and provides a full-scale output of 16380.
- 3 Bit Shift –This setting counts by 8's and provides a full-scale output of 32760.
- 4 Bit Shift – This setting is the least accurate, but allows the largest number. 4 Bit Shift counts by 16's and provides a full-scale output of 65520.

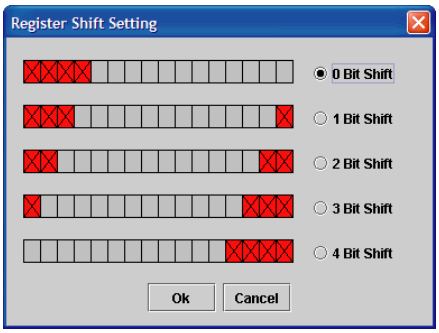
Register shifting is optional. You can skip this procedure if the default setting (0 Bit Shift) is sufficient (Note that the default setting will work with all DFS and OCS equipment). If you require a setting other than the default, it can be chosen before or after the mapping process. However, register (bit) shifting *must* be done before the configurations are installed on the PLC033. For more details on the radio mapping process, see the next section "Map Registers to DFS Function Modules."

To select a shift register setting for a point:

1. After adding a module to the DFS Radio Map tree (see steps 1 and 2 in the next section), right click the desired point in the tree and select Register Shift.

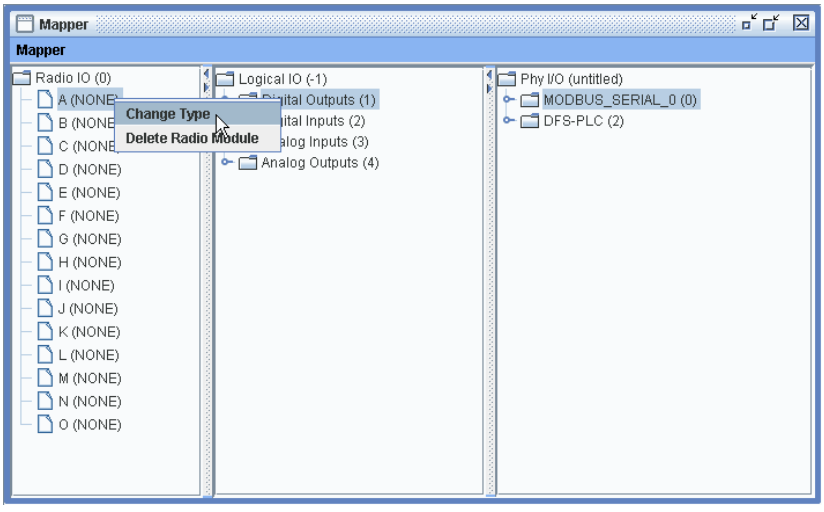


2. Select the register shift setting to use for this point and click OK.

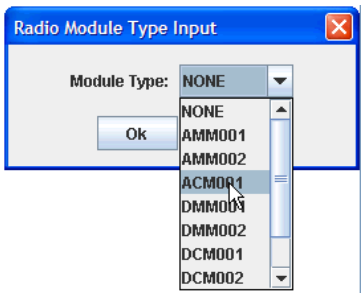


MAP REGISTERS TO DFS FUNCTION MODULES

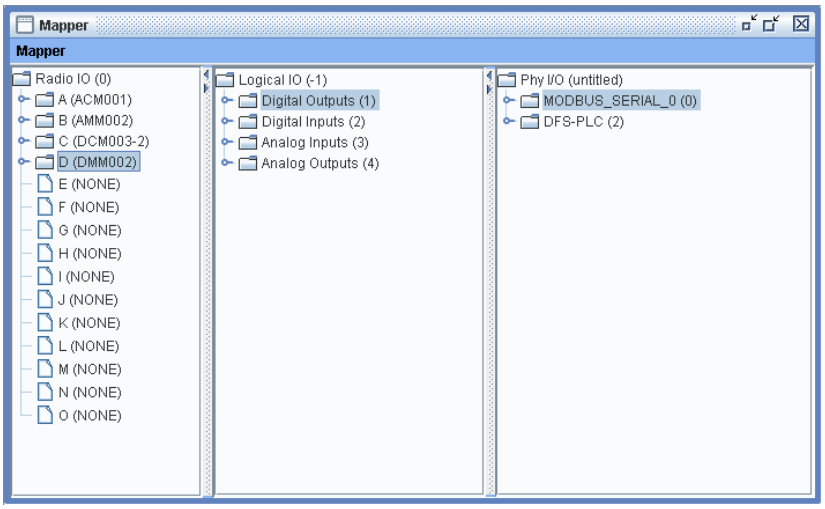
1. In the Radio I/O tree (left panel), select an empty module node and right click. Select **Change Type** from the pop-up menu.



2. In the **Radio Module Type Input** dialog box, select the desired module from the drop-down list. The node changes and expands to display the selected module and its corresponding I/O.



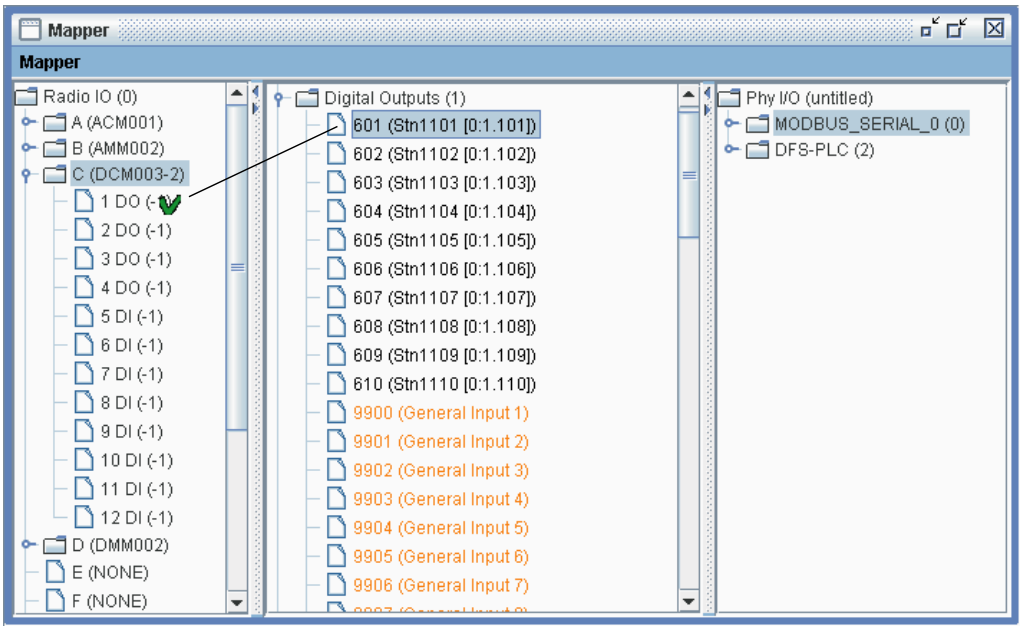
3. Continue adding modules until you have a sufficient number – in the right combinations – to accommodate the I/O that will be mapped.



4. In the Logical I/O tree, expand the I/O type you want to map first (for example, Digital Outputs), so all of the I/O is visible.

(continued on next page…)

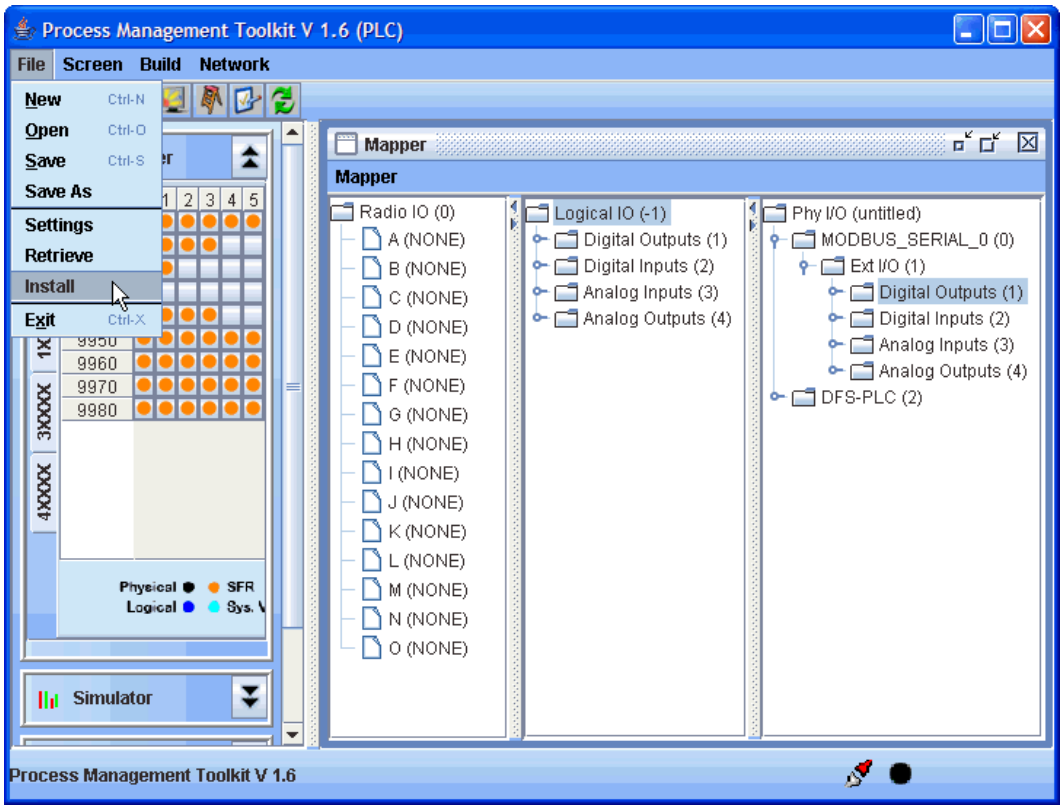
5. In the left panel (Radio I/O), expand the module type you want to map to so that all of its points are visible. Drag a logical register from the middle panel *directly* to the desired module point.



6. When all of the I/O has been mapped, save the project and install the configurations (see next section).

INSTALL CONFIGURATIONS ON THE PLC

When you are finished configuring and mapping I/O, you can install the configurations on the PLC by choosing **Install** from the **File** menu.



Observe the PLC's status LED. As it processes the configurations, the blinking pattern will be 3 quick flashes followed by a pause. The status LED will then turn off for a few seconds. When the installation is complete, the status LED will return to its normal blinking pattern (slowly turns on and then off).

Note that this only installs the I/O configurations and custom screens. Ladders are installed using a separate process.

RETRIEVE INFORMATION FROM AN EXISTING PLC

The Retrieve command enables you to retrieve information from an existing PLC. The information downloaded includes all of the PLC's configured I/O, register and DFS radio maps, ladders, and screen definitions.

This tool is useful if you are installing a new PLC that has a similar set up to an existing one. Instead of creating all of the I/O, maps, screens, and ladder from scratch, you can simply download the information, make any necessary changes, and transfer it to the new PLC. It is also useful for troubleshooting purposes. If the person doing the troubleshooting doesn't have the original project file, they can simply download it to their laptop.

1. Open PMT. Create and save a new project.
2. Choose **Settings** from the **File** menu. In the **Target** field, enter the IP address of the PLC you are retrieving information from.
3. Choose **Retrieve** from the **File** menu to begin downloading the PLC's configurations. When the **File Retrieve** window opens, click **Yes** to replace all local records with remote records.
4. Click **Start Transfer**. The **Transfer Progress** window shows the progression of the download.

   NOTE: If any of the screens contain images that are not part of the default package, for example a satellite image of your plant, those images will not be displayed unless they are in the PMT images folder on your computer. When screens are transferred to the PLC, only the screen definition is transferred. The images reside on the local machine. Any images that are not part of PMT's default image library must be copied to PMT's images folder (usually C:\Program Files\PMT2\Images).
5. When retrieval is complete, save the project. If this project is going to be transferred to another PLC, make any desired changes to the network settings, I/O, maps, ladders, and screens and then transfer it to the new PLC.

LOGIC BUILDER OVERVIEW

Logic Builder is a user-friendly application that enables you to construct "ladder logic"-style programs that run on the PLC. Ladder logic is a graphical (symbols and text) language that is used to plan, maintain and control industrial systems. Each rung of the ladder (hence the name - ladder logic) is used to control a single output. In many instances, the objects in a ladder logic diagram look like symbols used in electrical schematics. (An example of a simple ladder logic diagram is shown below.)

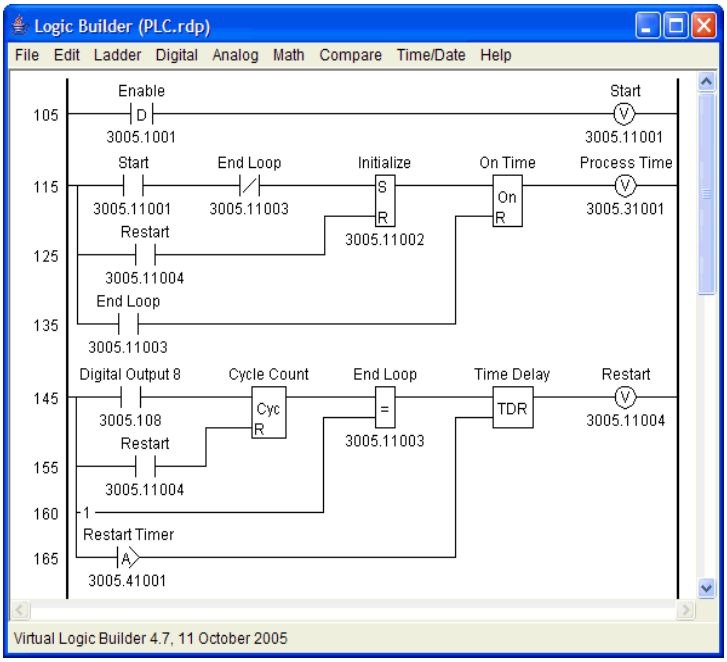


Figure 4-1: Sample Ladder Logic Diagram

The ladder logic diagrams – built using groups of rungs and branches – are actually programs you create to manage complex control functions. These graphical programs are converted into logical I/O and auto controls, and are continuously scanned by the system. The PLC033 is designed to complete one program scan per second. The speed of the scanning process enables you to have the most up-to-date information and allows you to react to situations quickly and efficiently.

In traditional ladder logic, the values that flow along rungs and branches are strictly logical – 0 or 1. DFS' Logic Builder provides the extra flexibility of allowing rungs and branches to hold numeric values, including inputs from analog registers and the results of math operations, such as Add and Maximum.

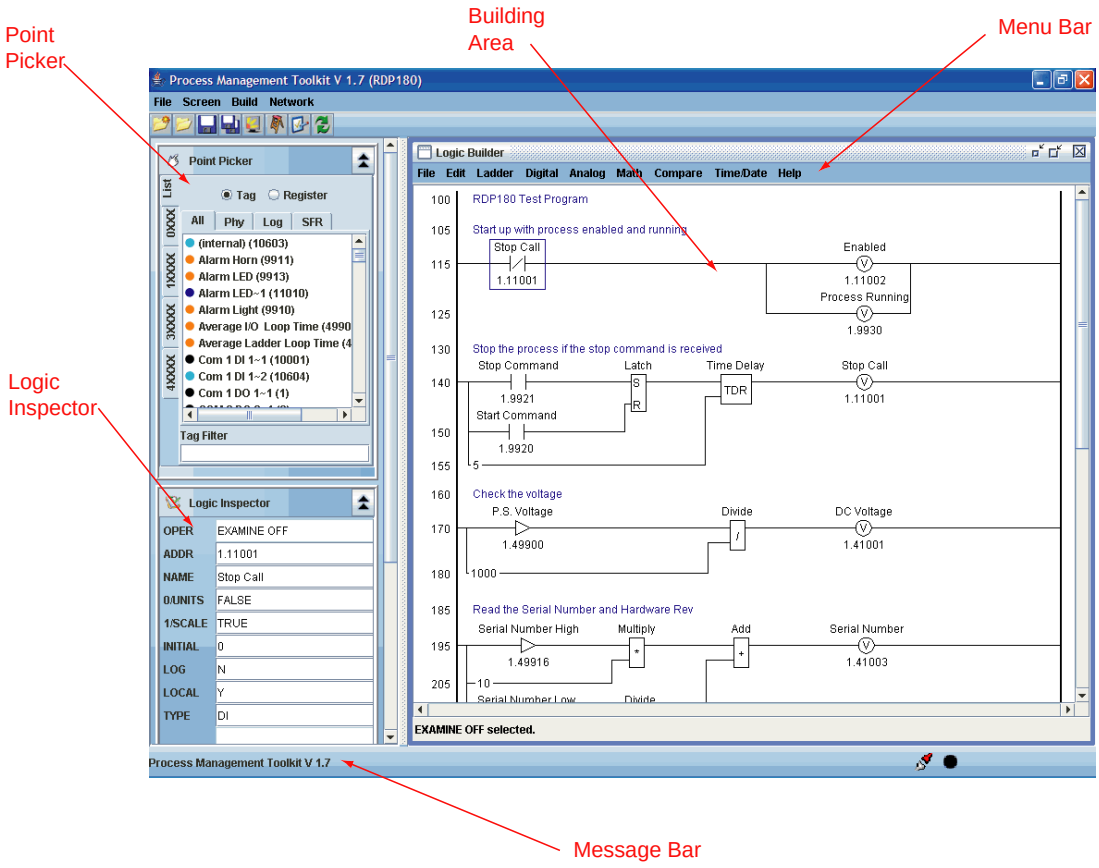
After a ladder logic diagram has been built and installed, you can use its logical I/O and auto controls in custom screens – graphical representations of your telemetry system that can be used to remotely monitor and control your system. (See “Chapter 5: Building User Interfaces” for more information on building custom screens.)

OPENING LOGIC BUILDER

To open Logic Builder, select Logic from the Build menu. Logic Builder opens in the right panel, and Logic Inspector [see “Editing Properties: Logic Inspector and Point Picker” on page 81] is added to the list of tools in the left panel.

LOGIC BUILDER INTERFACE

Before you begin using Logic Builder, it is helpful to familiarize yourself with the application's graphical user interface (GUI). The sections below describe the components that make up Logic Builder's GUI – its menus and menu commands, the building area, the message bar, the register selector, and Inspector.



THE LOGIC BUILDER GUI

The basic components of Logic Builder's GUI (shown on the previous page) are:

- **Menu Bar** – Features nine menus with commands. These commands are used to build, test, save and install your ladder logic diagrams.
- **Building Area** – Area where you build your ladder logic diagrams. When you start Logic Builder, an empty ladder - represented by two vertical rails, one each on the left and right side of the building area - is displayed.
- **Message Bar** – Displays messages, such as prompting messages and error messages, that provide you with additional information and instructions on Logic Builder functions.

Additionally, Point Picker and Logic Inspector appear in PMT's tool panel.

- **Point Picker** – Displays a list of available registers. The list can be sorted by tag name, register, or I/O type. See "Point Picker Basics" on page 21 for more information.
- **Logic Inspector** – Displays the selected object's parameters. For more information, see "Editing Properties: Logic Inspector and Point Picker" beginning on page 81.

MENUS

FILE

The File menu features the following commands:

- **New** – Create a new ladder logic diagram. If a diagram is already open, it is replaced with a new one and the logical address prefix is reset (see Settings, below).
- **Open File** – Import a ladder logic file (.vlb file) from another location (for example, from a Hyper SCADA Server). To open a PLC project file (.rdp file) that is stored on your computer, choose Open from the main console's File menu. The ladder logic program is a part of a larger project file that has an ".rdp" extension.
- **Save File** – Export the ladder logic file (.vlb file) to a location other than your computer (for example, to a Hyper SCADA Server). To save the PLC project file (.rdp file) to your computer, choose Save As or Save from the main console's File menu. The ladder logic program is a part of a larger project file that has an ".rdp" extension.
- **Settings** – Select the type of ladder and enter values for Temp Offset, PLC Station, and Default Resolution. Modbus must be selected when building a ladder for a PLC.
- **Change Station** – Change the station number used in the ladder, so it can be installed on another PLC. All references to the old station number are replaced with the new station number.
- **Install** – Write the ladder logic program to the PLC's temporary memory (i.e., the program is not retained if the PLC is reset). When a ladder is installed, any existing ladder program is replaced with the new one, and the PLC begins running the new program. Use Install when developing and debugging ladders. Use the Flash command to permanently record the program into the PLC's Flash memory. [**IMPORTANT**: Set point values are written to memory before an Install and are reset to those saved values when the install process is complete.]
- **Update Setpoints** – This command can be used when tuning a process algorithm. It copies the current values of the set points (typically analog input blocks) from the running process to the

Initial fields of the input blocks. This command prompts a PLC reset. After the set points have been updated and the PLC has reset, the saved ladder reflects the tuned state of the process.

- **Flash** – Permanently record the ladder logic program into the PLC's Flash memory. This command prompts a PLC reset (reboot). Use this command when you are satisfied with your changes, and you want the current program stored for use after future resets. When developing and debugging ladders, use the Install command, which places the program in temporary memory. [**IMPORTANT**: Set point values are written to memory before an Flash and are reset to those saved values when the flash process is complete.]
- **Uninstall** – Removes the ladder logic program from the PLC's temporary/Flash memory. This doesn't have any affect on the file stored on your computer.
- **Print** – Print the current ladder logic diagram.
- **Close** – Close Logic Builder.

EDIT

The Edit menu features the following commands:

- **Undo** – Reverse the last editing operation. Will *redo* the last editing operation if it was an undo.
- **Cut** – Remove the selected object(s) and place them on the clipboard.
- **Copy** – Copy the selected object(s) and place them on the clipboard.
- **Paste** – Insert the clipboard contents at the cursor location.
- **Delete** – Permanently remove the selected object(s).
- **Control (Force)** – Change the value of the selected object. In Simulate mode, you can change the value of *any* object that produces a value. Using Control (Force) in Simulate mode allows you to test your ladder logic without affecting a real device. For ladders that have been installed or flashed, you can only control Digital Inputs, Analog Inputs, and Momentary Inputs. This would typically be done to adjust set points or start/stop something.

LADDER

The Ladder menu features the following commands:

- **Rung** – Insert a new rung at the top of the diagram or after the selected object.
- **Branch** – Insert a new branch at the start of the selected rung or branch.
- **Comment** – Insert a comment at the top of the diagram or between rungs.
- **Go To** – Jump to a specified line number.
- **Find** – Search for a specific address or description.
- **Find Again** – Repeat the most recent find operation.
- **Cross Reference** – Enable or disable the display of reference line numbers under Out objects.
- **Check** – Check the current diagram for completeness and good addresses.
- **Refresh** – Sample (once) the values of the ladder's objects and display all values and colors.
- **Animate** – Repeatedly sample the values of the ladder's objects and display all values and colors.
- **Simulate** – Developing and debugging tool that allows you to simulate a ladder program's logic. In Simulate mode, the Control (Force) command can be used on most objects in a ladder to test the logic without affecting real hardware.
- **Stop** – Stop an animated display. Choosing an editing command also causes the animation to stop.

DIGITAL

The Digital menu features the following commands:

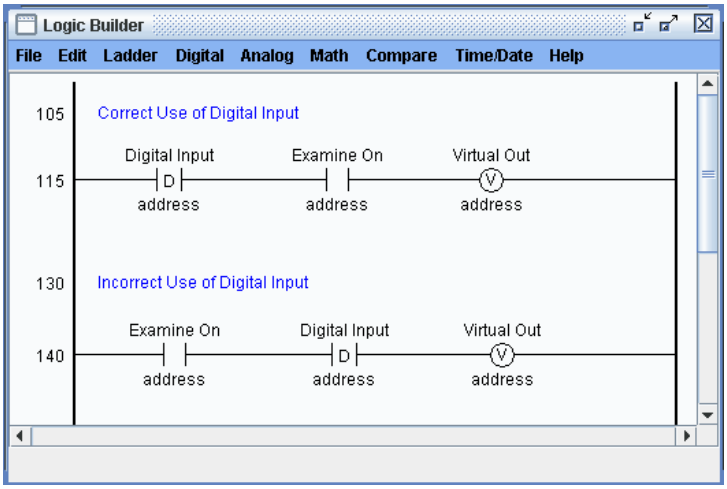
- **Examine On** – Retrieve (sample) the value of an existing Digital register and "*and*" it with the input value.
- **Examine Off** – Similar to Examine On, but the retrieved (sampled) value is inverted with a system-generated "*not*" equation.
- **Digital Input** [1] – Create a logical user digital input (can be set to 0 or 1) that can be used to manually control a register.
- **Momentary Input** [2] – Create a manual logical user input that changes state for a specified number of seconds.
- **Out** – Create an auto control that outputs from the ladder to a control register.
- **Out Not** – Create an auto control with the invert parameter set.
- **Virtual Out** – Create a simulated output relay for reference elsewhere in ladder logic.
- **Latch** – Set or reset the output when either input is true; otherwise, keep the previous value.
- **One-Shot** [3] **(DIFU, i.e., Differentiate Up)** – Momentarily turn on an output when the state of a specified input changes from off to on. The output remains on (true) for one program scan. [**NOTE**: Changing the One-Shot (DIFU)'s Initial value to 1, causes it to behave as a DIFD (Differentiate Down), i.e., the output remains off (false) for one program scan.]
- **Time Delay** [3] – Use a configurable timer to turn an output on, provided the output's rung is true. When the timer expires, the output is turned on. If the rung goes false anytime before the timer expires, the timer is reset. A constant is typically used to set the timer's maximum time, but any numeric value, for example an analog input (set point), could be used. To use a Time Delay to turn an output off, set its INITIAL value to 1.
- **Retentive Timer** [3] – Similar to a Time Delay, except the timer isn't reset when the rung goes false. The value of the accrued time is retained. When the rung goes true again, the object will begin accruing time starting at the retained value. This object has a third input that can be used to reset the timer. The reset should be a digital value (on to reset, off to run). Use a Momentary Input to allow a user to manually reset the timer. If the third input isn't used, the timer will automatically reset at midnight. To use a Retentive Timer to turn an output off, set its INITIAL value to 1.
- **4-State** – Combine two logical inputs to create a numeric output with a value from 0 to 3.
- **Cycle** – Count the number of times a logical input has turned on.
- **On Time** – Calculate the number of seconds a logical input is on.

NOTES:

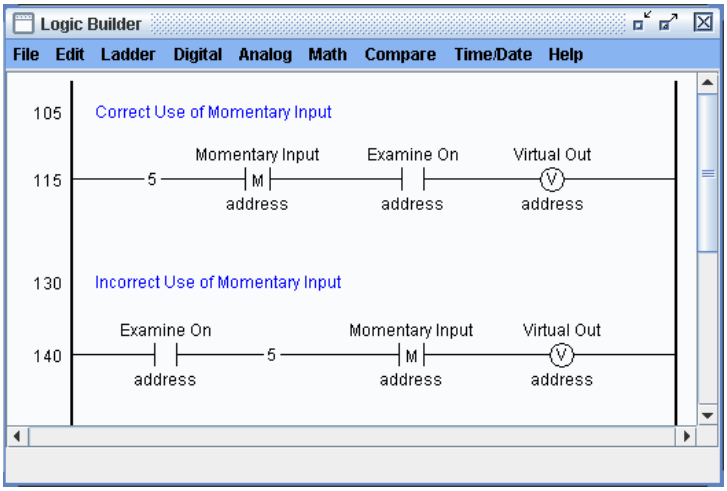
1. The **Digital Input** object does not use an input therefore it can't be used in AND functions unless it is the first object on the rung (see "Digital Input Example" on next page)
2. The **Momentary Input** object uses a constant or analog value as an input therefore it can't be used in AND functions unless it is the first object on the rung (see "Momentary Input Example" on next page).
3. **One-Shot (DIFU)**, **Time Delay** and **Retentive Timer** use their INITIAL parameter to determine what result to produce. Setting INITIAL to 0 (default setting), results in a change from 0 to 1. Changing INITIAL to 1, results in a DIFD, Time Delay Off or Retentive Timer Off command. If you

are familiar with virtual point equations, note that the virtual equation Oneshot is named Momentary Input in Logic Builder. Logic Builder's One-Shot (DIFU) is another name for the DIFU virtual equation.

Digital Input Example:



Momentary Input Example:



ANALOG

The Analog menu features the following commands:

- **Constant** – Replace the value of the rung or branch with the specified constant value.
- **Examine Analog** – Replace the value of the rung or branch with the value of an existing analog register.
- **Analog Input** – Create a virtual user analog input (can be set to any value) that can be used to manually control a register.
- **Deadband** – Used in ladders to prevent small input changes from affecting an output. Typically this output is connected to an auto-control.

- **Analog Out** – Create an auto control that outputs from the ladder logic diagram to an analog control register.
- **Virtual Out** – Create a simulated output relay for reference elsewhere in ladder logic.
- **Move** – Copy (move) the second input value while the first input is true; otherwise, keep the previous value.
- **Selector** – Output the second input value if the first input is true; otherwise, output the third input value.
- **Total** – Sum the absolute value of a numeric input each time a new value arrives. Optionally, this object can be reset at a specified time. If the reset is left blank, it is automatically reset at the start of each day (midnight).
- **Flow** – Convert a per-minute analog input into a totalized measurement. Optionally, this object can be reset at a specified time. If the reset is left blank, it is automatically reset at the start of each day (midnight).
- **PID** – Create a process control strategy using a process variable and a set point input, along with gain, integral, derivative, and integral limit ("wind-up") parameters.

NOTE: An Analog Virtual Out is similar to a Digital Virtual Out except that the 0/Units and 1/Scale parameters are numeric values instead of On and Off.

MATH

The Math menu features the following commands:

- **Add** – Output the sum of two inputs.
- **Subtract** – Output the difference between the value of the first input and the value of the second input.
- **Multiply** – Output the product that results when the first input is multiplied by the second input.
- **Divide** – Output the quotient that results when the first input's value is divided by the second input's value. Output 0 (zero) if the value of one or both inputs is 0.
- **Modulus** – Output the modulus (remainder) that results when the first input's value is divided by the second input's value. Output 0 (zero) if there is no remainder.
- **Square Root** – Output the square root of an input.
- **Minimum** – Output the lesser of two input values.
- **Maximum** – Output the greater of two input values.

COMPARE

The Compare menu features the following commands:

- **Equal** – Output 1 if both inputs have the same value; otherwise, output 0.
- **Even** – Output 1 if the input's value is even; otherwise, output 0.
- **Odd** – Output 1 if the input's value is odd; otherwise, output 0.
- **Less** – Output 1 if the first input's value is less than the second input's value; otherwise, output 0.
- **Greater** – Output 1 if the first input's value is greater than the second input's value; otherwise, output 0.
- **Less or Equal** – Output 1 if the first input's value is less than or equal to the second input's value; otherwise, output 0.

- **Greater or Equal** – Output 1 if the first input's value is greater than or equal to the second input's value; otherwise, output 0.

TIME/DATE

The Time/Date menu features the following commands:

- **Second** – Replace the value of the rung or branch with the number of seconds that have passed in the current minute (0-59).
- **Minute** – Replace the value of the rung or branch with the number of minutes that have passed in the current hour (0-59).
- **Hour** – Replace the value of the rung or branch with the number of hours that have passed since midnight (0-23).
- **Time of Day** – Replace the value of the rung or branch with the number of seconds that have passed since midnight (0-86399).
- **Day of Month** – Replace the value of the rung or branch with the day of the month (1-31).
- **Month of Year** – Replace the value of the rung or branch with the month of the year (1-12).
- **Year** – Replace the value of the rung or branch with the current year.
- **Day of Week** – Replace the value of the rung or branch with the day of the week (1-7, where Sunday = 1).
- **Day of Year** – Replace the value of the rung or branch with the day of the year (1-366).
- **Week of Year** – Replace the value of the rung or branch with the week of the year (1-53).

HELP

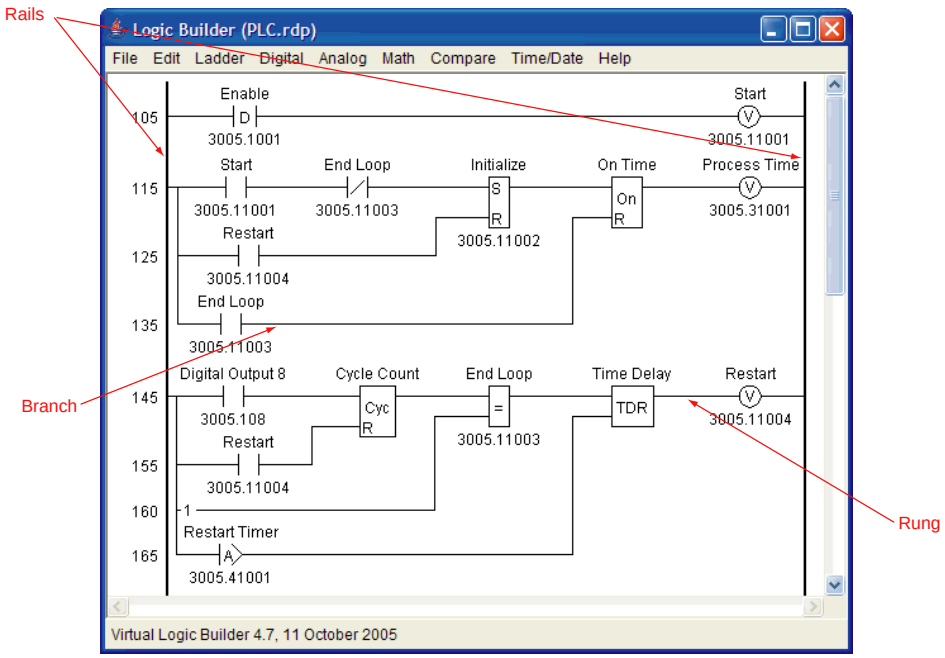
The Help menu features the following command:

- **About** – Display Logic Builder’s version number and release date in the Message bar.

WORKING WITH LADDERS

THE LADDER

When you start Logic Builder, you are presented with an empty ladder. The two vertical lines that appear on either side of the Logic Builder window are referred to as "rails." The rails and the horizontal lines - "rungs" and "branches" - that you add to the ladder logic diagram form a structure that resembles a ladder, hence the name ladder logic.



IMPLIED FUNCTIONS: AND AND OR

AND FUNCTION

The *AND* function is implied in ladder logic (i.e., there isn't an “And” object or operator that you insert in a ladder logic diagram). Examine objects (Examine On, Examine Off and Examine Analog) are used to construct *AND* statements. By default, the left edge of each rung has the value 1 (one). When you place an Examine on an empty rung, the output (the portion of the rung extending from the right side of the Examine) is equal to the current value of the Examine. Placing another Examine on the rung creates an *AND* statement. If the first input *and* the second input are both true, then their output is true. All other combinations result in a false, or 0, output value. See the table below.

Table 4-1: Output of AND Function

Value of First Input	Value of Second Input	Combined Output
1	0	0
0	1	0
0	0	0
1	1	1

AND functions as follows:

Sample the value of the first Examine (Pump 1 Fail) and the value of the second Examine (Pump 2 Fail). If both the first input *and* the second input are true (1), then the output is true (Pump Override's value becomes true). Otherwise, the output is false.

More than one Examine can be placed on a rung or branch, but the final output is true only if the value of *all* of the Examine objects is true.

THE OR FUNCTION

The *OR* function is similar to *AND* in that there isn't an object or operator that is inserted in a ladder logic diagram. Examine objects (Examine On, Examine Off and Examine Analog) and branches are used to construct *OR* statements. The first Examine and an output object are placed on an empty rung. A branch is placed on the diagram. The branch starts just before the first Examine and ends just before the output. Another Examine is placed on the branch. If the first input *or* the second input is true, then their output is true. If both inputs are false, the output will be false. See the table below.

Table 4-2: Output of OR Function

Value of First Input	Value of Second Input	Combined Output
1	0	1
0	1	1
0	0	0
1	1	1

OR functions as follows:

Sample the value of the first Examine and the value of the second Examine. If the first Examine *and/or* the second Examine is true (1), then the output is true. Otherwise, the output is false.

More than one branch and Examine can be placed on a rung, but the final output is true only if the value of *at lease one* of the Examine objects is true.

LADDER FILES

Ladder files can be saved in three ways:

- As Part of a PMT Project – When you save a PMT project file, the ladder's definition (temporary variables, object locations, etc.) becomes part of the project. There can be only one ladder per project.
- As a Separate Entity – A ladder's definition can be stored outside of a project as a .vlb file. This is done by selecting **Save File** from Logic Builder's **File** menu. To open a ladder saved as a .vlb file, select **Open File** from Logic Builder's **File** menu.
- On a PLC – When you perform an **Install** or **Flash**, a copy of the ladder is saved to the PLC. The new file always *completely* replaces the old one.

EDITING LOGICAL I/O AND AUTO CONTROLS

You must use Logic Builder to make edits to logical I/O or auto controls that were created in or generated by Logic Builder. Do *not* use I/O Builder to make changes to these registers. Edits made with I/O Builder do not affect your Logic Builder file. Any changes made using I/O Builder will not be reflected the next time you open your ladder logic diagram in Logic Builder and may adversely affect the program's function.

CREATING NEW LADDERS

To create a new ladder, choose New from the File menu.

Note that only one Logic Builder file can be open at a time. If a file with unsaved changes is already open and you attempt to create a new one, a dialog box with the message "[file name].vlb not saved. Discard changes?" appears. If you want to save the changes to the current file, click Cancel, save your changes, and then create a new file. If you don't want to save the changes, click Discard.

SETTING LADDER PROPERTIES

From the Settings dialog box (choose Settings from the File menu), you specify the type of device the ladder will be installed in (Modbus). You can also specify a Temp Offset, the PLC station number, and a default resolution.

LADDER TYPE

For a PLC033 or RDP, select Modbus.

TEMP OFFSET

This setting specifies the starting point where temporary variables should be written to the PLC's Logical Memory Map. Some ladder logic objects, for example, the if-then object Move, are temporary variables. Although they may be evaluated and produce a value every time the program is scanned, it isn't necessary to permanently store their value.

For example, if you select 500 for the offset, all temporary digital input variables will be written to the Logical Memory Map beginning at register 10500 and all temporary analog input variables would start at register 30500. By assigning real I/O and auto control variables to the lower registers, conflicts that would arise if the temporary variables were written to those assigned registers can be prevented.

PLC STATION

Enter the station number assigned to the PLC you are creating the ladder for. You can easily use the ladder for a PLC at another station by changing this setting to the number assigned to the other PLC's station.

DEFAULT RESOLUTION

This setting specifies whether calculations are done using integer arithmetic or floating point arithmetic. It is placed in the scale field of each new analog object you create.

- For integer arithmetic, enter a resolution of 1.
- For floating point arithmetic, enter a resolution less than 1 (e.g., .1 or .01). Keep in mind that floating point arithmetic may be slower.

SAVING AND OPENING

OPEN FILE LIMITS

Only one Logic Builder file can be open at a time. If a file with unsaved changes is already open and you attempt to open another one, a dialog box with the message "[file name].vlb not saved. Discard changes?" appears. If you want to save the changes to the current file, click Cancel, save your changes, and then open another file. If you don't want to save the changes, click Discard.

SAVE, INSTALL, AND FLASH

Saving a ladder using Logic Builder's **Save File** command only places a copy on your local computer; it does not install it on the PLC. Ladders saved this way use a .vlb extension. A .vlb file can be used as a template for other PLC's. Additionally, the ladder is saved as part of the main project file when you select **Save** from the main console's **File** menu. Each project can have only one ladder associated with it.

An installed ladder is saved in the PLC's temporary memory. An installed ladder is *not* retained if the PLC is reset or power is cycled. Install is useful when you are developing and testing your program.

Flash is used to install the ladder to the PLC's permanent Flash memory. A flashed ladder *is* retained in the PLC's Flash memory if the PLC is reset or power is cycled. Flash should only be used when you are satisfied with the results of your testing. Note that flashing a ladder causes the PLC to reboot.

IMPORTANT: Set point values are written to memory before an Install or Flash and are reset to those saved values when the install or flash process is complete.

DOCUMENTING

CROSS REFERENCES

Logic Builder's Cross References feature, used in conjunction with the Go To command, can help you easily find and jump to the location(s) where a specific output appears.

Cross References are displayed below all output objects and list the other line numbers where this particular address appears. You can use the Go To command to jump to the line numbers that are listed below the object.

Turn On Cross References

Choose Cross References from the Ladder menu. The Message bar displays "References are displayed below OUT objects" and a check mark appears beside the Ladder menu's Cross References command.

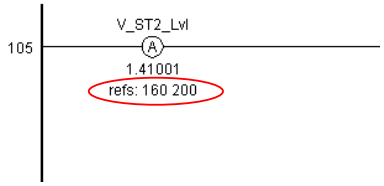


Figure 4-2, Cross References

Displayed below each output object is "refs: [#]," where [#] is the line number where this object's address is referenced. If the address appears on multiple lines, all the line numbers are listed. If an output object's address doesn't appear anywhere else in the diagram, nothing is listed below it.

Turn Off Cross References

Choose Cross References from the Ladder menu. The Message bar displays "Cross References are now off." The check mark next to the Cross References command and the line numbers listed below the output objects disappear.

COMMENTS

Comments can be added to your ladder logic diagrams to provide insight into the function of a particular section of logic.

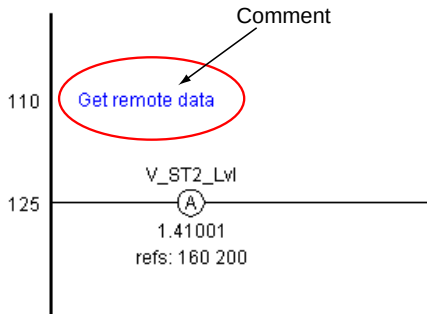


Figure 4-3, Comments

NOTES:

- Comments are left-aligned and can be placed above or below a rung; they can't be inserted in the middle of a rung.
- To place a comment at the *top* of your ladder logic diagram, verify that nothing is selected (rung, object or operator), and then add your comment.
- To place a comment *below* a particular rung, select the last object on the rung and then add your comment.

Add a COMMENT

1. Determine where you want to place the comment (see the notes above) and then choose Comment from the Ladder menu.
2. Add text to a comment using either of the following methods:
   - Select the comment and press Tab, or simply double click the comment. The default or existing text is highlighted. You can delete this text (press Backspace key) and enter your own text, or simply type over the highlighted text.
   - In Inspector, type the desired text in the NAME box and press Enter.
3. Assign a line number to a comment using either of the following methods:
   - Select the comment and press Tab twice. The default or existing line number is highlighted. You can delete this line number (press Backspace key) and enter your own, or simply type over the highlighted number.
   - In Inspector, type the desired line number in the ADDR box and press Enter

TESTING AND TROUBLESHOOTING LADDERS

CHECK

The Check command examines your ladder logic diagrams and returns error messages if Logic Builder encounters any problems as it scans the diagram from beginning to end.

Check inspects the diagram for:

- Completeness – Do all objects that require addresses or inputs have them?
- Valid addresses – Are all assigned object addresses configured; do any system-generated logical addresses conflict with existing ones?

If Logic Builder finds any errors in the diagram, an error message displays in the Message bar and the object in question is surrounded by a blue line. After you correct an error, you must check the diagram again to see if any others exist. Logic Builder displays the error messages one-by-one in the order in which they occur; it does not display *all* the errors at once.

Additionally, the ladder must be checked before any logical I/O created in the ladder will appear in the Register Selector list.

IMPORTANT: Always check your ladder logic diagrams before saving and installing them.

CONTROL (FORCE)

Control (Force) enables you to change the value or state of a single object.

In Simulate mode, you can control/force any object that produces a value without affecting a real device: Simulate works entirely within your computer. Using Control (Force) in Simulate mode enables you to test how your ladder reacts to different input conditions without having to physically create those conditions.

When not in Simulate mode, you can only control/force Digital Inputs, Analog Inputs, and Momentary Inputs. These control inputs are typically used to adjust set points or start/stop something.

Control (Force) an object

1. Select the object you want to control.
2. Choose Control (Force) from the Edit menu. The Control dialog box appears and the Message bar displays "Enter the new value for [object address]." Where [object address] is the logical address of the object you want to control.
3. Enter the desired digital state (0 or 1) or analog value, and click OK. The Message bar displays "Setting control [object address] to [state or value]." Where [object address] is the logical address of the object you are controlling and [state or value] is the desired digital state/analog value. To exit the Control dialog box without changing the object's state or value, click Cancel.

After the object's state or value has been set as indicated in the Control dialog box, you can see the results of the Control (Force) by checking the ladder logic diagram.

REFRESH, ANIMATE, SIMULATE, AND STOP

Refresh and Animate are similar in that they fetch the current state or value of the registers contained in a ladder logic diagram and display them on the screen. In this way, you can see your ladder logic in action and determine if any changes need to be made.

The difference between Refresh and Animate is in how they retrieve (sample) the data.

Refresh is a static representation, like taking a snapshot. Refresh shows you what the states and values were when you clicked Refresh. Any changes that occur after the "snapshot" is taken are not reflected on the screen.

Animate is a live-action representation of your system. It reflects and displays any changes in state or value as they occur. It will continue to retrieve data until you either select the Stop command or select an editing command. With Animate, you can affect the logic – as it executes – by using the Control (Force) command to control the ladder's input objects (Digital Input, Analog Input, Momentary Input). To use Animate, the ladder must be Installed on the PLC.

Simulate is useful when developing and debugging programs, because it allows you to simulate a ladder program's logic without installing the ladder and affecting real hardware. When Simulate is turned on, you can use the Control (Force) command to change the value of any object that produces a value and see how it affects the operation of the ladder. Simulate enables you to determine if the results are what you expected and desired before installing the ladder program on the PLC.

The Refresh, Animate, Simulate, and Stop commands can be found in the Ladder menu.

VIEWING SYSTEM ASSIGNED REGISTERS IN LOGIC BUILDER

Not all objects require you to manually assign them a specific register. Registers for objects such as the math objects Add and Subtract are assigned by the system. When troubleshooting ladders it is sometimes helpful to see the address of such objects.

To view the system assigned register of an object, right-click the object. The assigned register is displayed next to the Output Address label.

INSTALLING AND UNINSTALLING

INSTALLING

There are two methods for installing a ladder on the PLC. While you are testing your logic, you can use the **Install** command to place the ladder in the PLC's temporary memory. When the ladder has been thoroughly tested, you can use the **Flash** command to place it in the PLC's permanent (flash) memory.

When a ladder is installed, the program is written to the PLC's temporary memory (i.e., the program is not retained if the PLC is reset). Any ladder stored in the temporary memory location is replaced with the new one, and the PLC begins running the new program.

When a ladder is "flashed," it is permanently recorded into the PLC's Flash memory. This command prompts a PLC reset (reboot). Use this command when you are satisfied with your logic, and you want the current program stored for use after future resets.

Both of these commands – **Install** and **Flash** – can be found in Logic Builder's **File** menu.

1. Open Logic Builder.
2. Choose **Install** (or **Flash**) from the **File** menu. The Message bar flashes "updating..." while the ladder logic diagram is being installed. If any errors are encountered during the installation process, they are displayed in the Message bar.
3. When Logic Builder is finished installing (flashing) the ladder, the Message bar displays "installed successfully." If the ladder was flashed, the PLC will reboot when the flash is complete.

UNINSTALL

Uninstall removes all the logical registers generated by a ladder logic diagram.

Uninstalling a ladder logic diagram only removes it from the PLC's temporary/Flash memory. It has no affect on the project file or the .vlb file stored on your computer.

1. Open Logic Builder.
2. Choose **Uninstall** from the **File** menu. If any errors are encountered during the uninstall process, they are displayed in the Message bar.
3. When Logic Builder is finished uninstalling the ladder logic diagram, the Message bar displays "uninstalled successfully."

PRINTING

To print your ladder logic diagram:

1. Choose Print from the File menu.
2. Select a printer from the Print dialog box and click OK.

WORKING WITH LADDER OBJECTS AND OPERATORS

OBJECT ADDRESS REQUIREMENTS

- Examine and Out objects must be assigned to an existing register in the PLC's Logical Memory Map.
- Input, Momentary and Virtual Out objects must be assigned to an unused register in the PLC's Logical Memory Map.
- Registers for all other objects are assigned by the system when necessary. It isn't necessary to manually assign them to a register. To view the system assigned register of an object, right-click the object. The assigned register is displayed next to the Output Address label. This is a useful troubleshooting tool. By simulating or animating the ladder, selecting the desired system assigned register in the Point Picker (see page 21), and viewing the register's value in the Simulator (see page 28) you can determine if the ladder is behaving properly.

ADDING COMPONENTS

Keep in mind that when a new component is added to a ladder logic diagram, the new component is always placed to the RIGHT of the selected component. You need to plan ahead when designing ladder logic diagrams to ensure that components are added in the correct order.

RUNG

The first step in building a ladder logic diagram is to add a rung. Rungs form the base of all ladder logic diagrams; they hold all of the objects and operators that are used to create the logic.

Choose Rung from the Ladder menu. A rung appears at the top of the diagram and the Message bar displays "RUNG created." The rung is surrounded by bright blue lines to show that it is selected. The number to the left is the line number where the rung is located.

To deselect the rung, move your mouse pointer to another position on the diagram and click.

You can move the rung up and down on the rails by selecting the rung, holding down the left mouse button (the pointer becomes a four-headed arrow) and dragging the rung to another location.

After inserting a rung, choose an object or operator from the Digital, Analog, Math, Compare, or Time/Date menu.

To add additional rungs to your ladder logic diagram:

- Below the existing rung – Select the last component on the existing rung and choose Rung from the Ladder menu. The new rung is inserted directly below the existing rung.
- At the top of the diagram – Verify that no components are selected and choose Rung from the Ladder menu. The new rung is inserted at the very top of the ladder logic diagram.

BRANCH

Branches are an integral part of ladder logic diagrams. Branches are used to build *OR* functions into logic or to make use of some logic to the left of the branch several times. They hold secondary inputs from objects and operators, such as Move and Less Or Equal that require more than one input to perform their function. Other objects, such as Time Delay and Cycle Count, need a branch that tells them how long to delay sending output or when they should reset.

Add a branch:

1. Add the objects that are to be the starting point and ending point for the branch.
2. Select the rung or branch that holds the starting point and choose Branch from the Ladder menu. The beginning of the branch appears to the left of the starting point and is connected to the rung or branch, and the mouse pointer becomes a cross-hair. Drag the cross-hair to the ending point and click. The complete branch appears on the diagram.

NOTES:

- For consistent results, always start a branch to the left of the starting point.
- If you try to connect a branch to something other than a rung or branch, or an object/operator such as Less Or Equal, the Message bar displays "cannot insert BRANCH here."

OBJECTS AND OPERATORS

To add components (other than a rung) to a ladder logic diagram, you must first select a component – rung, branch, object or operator. If you don't select a component, the Message bar displays "select an object first."

When new components are added, they are placed to the *right* of the selected object.

If you select an object, such as Examine Analog, and add another component, such as Virtual Out, Virtual Out is inserted to the right of Examine Analog.

If you select an empty rung or branch, the new component is added directly onto the selected rung or branch.

If you select a rung or branch that already contains components, the new component is added to the very beginning of the rung or branch.

EDITING PROPERTIES: LOGIC INSPECTOR AND POINT PICKER

LOGIC INSPECTOR

After adding an object to your ladder, you must define the object's parameters, including its register and user-defined name using Logic Inspector. Logic Inspector can be found on the left side of the main PMT window. It opens and remains docked in the tools panel whenever Logic Builder is opened.

Select the object and type the register number and I/O name, and any other necessary information, into Logic Inspector's fields.

Logic Inspector displays the selected object's parameters. Logic Inspector's fields change depending upon the type of object selected. See Logic Inspector Field Definitions in Table 4-3: Logic Inspector Field Definitions (page 83) for more information on each parameter.

Logic Inspector's left column lists the field names and its right column shows the field's current value. Some input fields, for example, OPER are not editable.

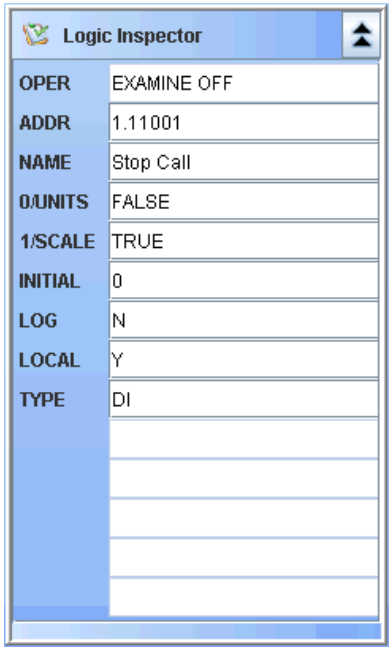


Figure 4-4, Logic Inspector

POINT PICKER

Point Picker, located in the left panel of PMT, can be used to choose the register that you want to associate with an object.

3. Select the object (blue box appears around object).
4. Locate the register in Point Picker and double-click it. The ADDR field in Logic Inspector is updated to show the selected register; the tag name associated with the register will appear in the NAME field.

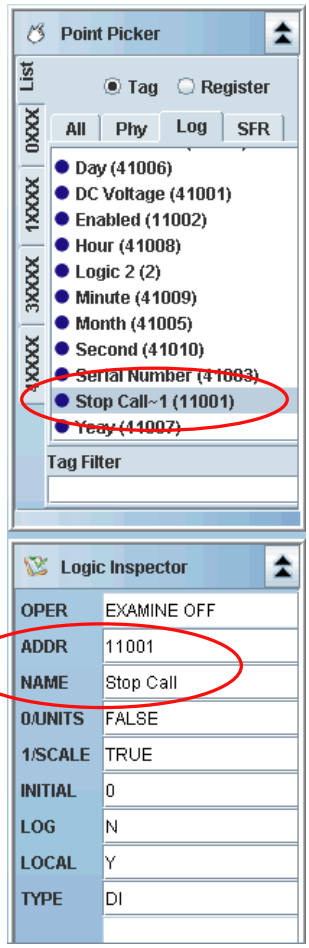


Figure 4-5, Selecting a Register for a Ladder Object

Shortcut: If you want to quickly change an object's name or register without going to Inspector:

1. Double click the object. The ADDR field becomes active.
2. With the default or existing address highlighted, type the new address and then press Enter. The Name field becomes active.
3. With the default or existing name highlighted, type the new name and press Enter.

Table 4-3: Logic Inspector Field Definitions

Field Name	Description
OPER	Name of selected operator. This field cannot be edited; it is for reference purposes only.
ADDR	Register to associate with selected object. For control objects, ADDR would be the register of the destination - the register to which this object's output is sent.
NAME	Text string (up to 20 characters) that describes, or names, the selected object. This field *cannot* be left blank.
0/UNITS	Text string. For digital registers, this is the label to display during the 0 state. For analog registers, this is the unit of measurement (e.g., Degrees, PSI, etc.)
1/SCALE	Text string. For digital registers, this is the label to display during 1 state. For analog registers, this is the resolution, or minimum change to be reported. Enables you to set the number of decimal places.
LOW ENG	Used with auto control objects. This is the low threshold, or control level, in engineering units.
LOW RAW	Used with auto control objects. This is the low threshold, or control level, in raw values.
HIGH ENG	Used with auto control objects. This is the high threshold, or control level, in engineering units.
HIGH RAW	Used with auto control objects. This is the high threshold, or control level, in raw values.
INITIAL	Float decimal. Value to which this operator should be set at startup.
LOG	Not used at this time.
LOCAL	Y or N. Enabling Local confines this object to the PLC; it isn't broadcast to the rest of HT3. Disabling Local for objects that are only used in equations (are not "real" I/O) helps to conserve disk space.
TYPE	Typical I/O type for the selected object. This is for informational purposes only; the field cannot be edited.

SELECTING AND MOVING

SELECTING AN OBJECT

Before an object can be manipulated or edited in any way, you must first select it. Selecting an object is simply a matter of clicking it. When an object has been selected it is surrounded by a bright blue line and the Message bar displays "[object name] selected."

SELECTING MULTIPLE OBJECTS

It is also possible to select multiple objects so that an action can be simultaneously performed on the group. This is useful when you want to copy a group of objects to a new location or need to delete a group of objects. Instead of selecting and copying/deleting each item, you select all the objects and then perform the function.

To select a group of objects: click on any object in the group, then hold down the Shift key and click on the second, third, fourth, etc. As you click on objects, the Message bar displays "1 objects

selected," "2 objects selected," etc. When you have finished selecting objects, perform the desired command.

DRAGGING (MOVING) OBJECTS

You can move rungs and branches vertically by selecting the rung or branch, holding down the left mouse button, and dragging it up or down. For finer control of movement, select the rung or branch and use the arrow keys on the keyboard to move it up or down.

You can use the same methods to move other ladder logic objects, except that other objects can only be moved horizontally along the rung or branch they are positioned on.

RE-USING A VALUE

If you need to fetch, or sample, the value of an Input, Momentary or Virtual Out more than once in the current ladder logic diagram or in another ladder logic diagram, assign it to a unique unused register in the PLC's Logical Memory Map. Use Examine to fetch the object's value when it is needed.

TIMER AND COUNTER RESETS

- Time Delay automatically resets when the first input goes false.
- Retentive Timer has an optional reset input. The reset should be a digital value (On to reset, Off to run). Use a Momentary Input to allow a user to manually reset the timer. If the reset input is left blank, Retentive Timer resets itself at midnight.
- Cycle, On Time, Total and Flow have optional reset inputs. If the reset input is left blank, the object will be reset at midnight.

RESUME PREVIOUS VALUE AFTER RESTART

To allow objects, such as Latch, to resume their previous value after a PLC restart, enter NULL in the object's Initial field (in Inspector).

EDITING TOOLS

Logic Builder features standard editing tools, such as Cut and Paste, as well as commands that enable you to quickly move around your ladder logic diagrams.

UNDO

The Undo command is used to reverse an editing action, such as Cut or Paste, or to remove an item that you mistakenly added to your ladder logic diagram. Undo only reverses the last action taken; you can't undo an action taken, for example, three steps ago.

To reverse the last action performed, choose Undo from the Edit menu. If Undo can't reverse the last action, the Message bar displays "UNDO not available."

You can also reverse an undo (perform a redo) by simply choosing Undo from the Edit menu again.

CUT, COPY AND PASTE

The Edit menu's Cut, Copy and Paste commands can be used to perform such actions as moving an object from one location to another or making a duplicate of an object. Groups of objects can be cut or copied, and pasted by selecting multiple objects.

To cut or copy an object and paste it in a new location:

1. Select the object and choose Cut or Copy from the Edit menu.
2. Select where you want to insert the object and choose Paste from the Edit menu. (Review Working with Ladder Objects and Operators: Adding Components beginning on page 79 for more information on how Logic Builder determines where an object is placed.)

DELETE

The Delete command is used to delete ladder logic objects and operators.

To delete an object or operator:

1. Select the object to be deleted.
2. Choose Delete from the Edit menu.

When deleting objects that are related, or connected, to one another, you must delete the objects in the reverse order in which they were added. For example, your ladder logic diagram has an Equal operator whose second input is a branch that contains a Constant.

These objects were added in the following order: Equal, Branch, Constant. The objects must be deleted in the exact opposite order: Constant, Branch, Equal operator. If you attempt to delete the branch before the Constant, the Message bar displays "Error, related object(s) are not selected."

You can also select multiple objects and delete them simultaneously. See Selecting Multiple Objects on page 83 for more information.

GO TO

Go To enables you to quickly move from your current location in a ladder logic diagram to a specified line number. Go To, used in conjunction with Cross References, can be used to jump to line numbers where particular output object addresses appear.

1. Choose Go To from the Ladder menu. The Go To dialog box appears.
2. Enter the line number you want to jump to and click OK. Logic Builder jumps to the desired line number. To exit the dialog box without going to a new location, click Cancel.

FIND AND FIND AGAIN

Use the Find and Find Again commands to search your ladder logic diagrams for occurrences of a particular register or description. For example, to locate each object assigned to register 30001 or includes the word Pump in its name.

Find is case sensitive. For this reason, you should be consistent in how you name your objects. Determine a naming convention (all caps, all lowercase, title case, etc.) and stick with it. Being consistent will save you time and frustration when you are trying to locate an object in a ladder logic diagram that has numerous rungs.

1. Choose Find from the Ladder menu. The Find dialog box appears.
2. Enter the character string you want to search for in the Find What String box.
3. Indicate Where you want to search. Click Address to search the ADDR field. Click Description to search the Name field.
4. Indicate How you want to search.
    - Click Equals to find a string that exactly matches your search string. [Ex: To find objects assigned to register 10001, type the full and exact register (10001) in the Find What String box.]
    - Click Contains to find a string that includes your search string. [Ex: Searching by Description for Well, finds Wet Well Level #1, Wet Well Level #2, etc.]
    - Click Starts With to find a string that begins with the same characters as your search string. [Ex: Searching by Address for 105 would return 10501, 10510, 10599, etc.]
    - Click Ends With to find a string that ends with the same characters as your search string. [Ex: Searching by Description for Level finds Low Level, Lag Level, Lead Level, etc.]
5. Click Find to begin your search. Logic Builder jumps to the first occurrence of the string. If the string is not found, the Message bar displays "not found." To exit the Find dialog box without searching for the string, click Cancel.

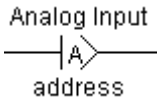
The **Find Again** command (choose Find Again from the Ladder menu) searches for the next occurrence of the most recent Find operation.

Using Find Again saves keystrokes when you need to find all occurrences of a particular search string or when the first search doesn't locate the exact object you are looking for. Find Again doesn't open the Find dialog box and doesn't require you to enter any information. If there is another occurrence of the string, Logic Builder jumps to that location. If there aren't any more occurrences of the string, the Message bar displays "not found."

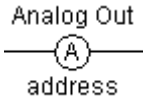
LADDER OBJECTS: DESCRIPTIONS AND PROPERTIES

ANALOG LADDER OBJECTS

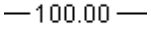
ANALOG INPUT

Description	Example
Create a logical analog input that can be set to any value and be used to control a register. Logic Builder features one Analog Input object. The typical use of this object is to provide some type of operator control or input *into* the logic. This object can be controlled by a remote event through the use of an auto control configured in HT3. Optionally, you can enable direct operator control by including the input in a custom screen. (see the ***PLC033 Installation and Operation Manual*** for more information). The Control command can be used on this object to test your logic. Note that Analog Input objects remain at the value set through the Control command until they are manually reset from the Control command. Analog Input objects must be assigned to an unused Register in the 40000 range.	

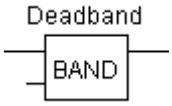
ANALOG OUT

Description	Example
Create an auto control record that outputs *from* the ladder logic diagram *to* an analog control. Logic Builder features one Analog output. This output is an auto control record that outputs *from* the ladder logic diagram *to* a specified analog control. Analog Out objects must be assigned to an unused Register in the 30000 range.	

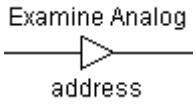
CONSTANT

Description	Example
Replace the value of the rung or branch with any specified constant. Constant can be assigned a numeric value that is no more than 10 characters long, including a decimal point. To assign a value to Constant, double click it. The default or existing value is highlighted and you can type in a new value. Press Enter when you are finished typing the new value.	

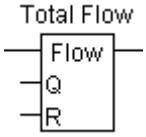
DEADBAND

Description	Example
The Deadband object can be used in ladders to prevent small input changes from affecting an output. Typically this output is connected to an auto-control to avoid excessive radio traffic. The second input to the Deadband object gives the size of the deadband. If the difference between the previous output and the first input is greater than the deadband, the first input is copied to the output. Otherwise the output remains unchanged. When using a Deadband with a PID object, set the Deadband to a small value. A large deadband will prevent the PID algorithm from working properly.	

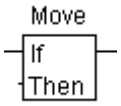
EXAMINE ANALOG

Description	Example
Replace the value of the rung or branch with the value of an existing Analog register. Logic Builder features one Analog Examine object that fetches the current value of a real or logical register. When Logic Builder reaches an Examine object, it retrieves its current value and assigns that value to the Examine object. Examine objects must be provided with an *existing* Register. This can be the register assigned to a real physical input/output or the register number assigned to a logical output that was configured somewhere else in the current ladder logic diagram.	

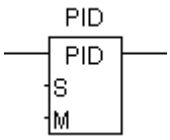
FLOW

Description	Example
Convert a per-minute analog input, such as gallons per minute, into a totalized measurement, such as total gallons. This object requires a qualifier address. If there is no qualifier address, a Constant with a value of one (1) must be placed on the qualifier input branch. This object also has an optional reset input that enables you to reset FLOW at a specified time. If the reset is left blank, FLOW is automatically reset at the start of each day (midnight).	

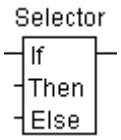
MOVE

Description	Example
Copy (move) the second input value while the first input is true; otherwise, keep the previous value. Move (and Selector) are used to construct conditional statements. They test the first input to see if it is true. If it is true, one course of action is taken. If it is not true, another action is taken. Move is an If-Then statement and requires two inputs. The first input is the test, or conditional statement. The second input states what happens when the first input is true. If the first input is true, then the second input's value becomes the output. If the first input is not true, then its value is retained as the output. Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.	

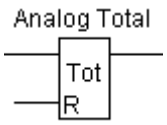
PID

Description	Example
The PID object uses the variables Gain, Integral, and Derivative to create a process control strategy that drives a measurable Process Variable towards a desired Setpoint. This section provides information on Logic Builder's PID object. More detailed information on using this object to implement PID control can be found at the end of this chapter in the section titled "PID Basics" beginning on page 98. PID features three inputs: Process Variable, Setpoint, and Manual (optional). • Process Variable (also called the process input) is the input you are trying to control (e.g., level, pressure, temperature, flow, or pH). • Setpoint is the desired value of the process variable. For example, the setpoint for a car's cruise control would be 70 mph if you wanted to maintain a speed of 70 mph. • Manual is optional. It is only necessary when the process variable has a manual switch that can be used to turn automatic control off and on. It is important to connect the manual switch to the PID object. When the two are connected and the process variable is in manual mode, the PID object is able to zero the integral part of the PID object to prevent accumulation of meaningless error values. The manual input can also be used to "zero" the accumulation of error when the process output is at its minimum and maximum value. To set the PID's parameters, select the object and open Inspector. **IMPORTANT**: The algorithm used by Logic Builder's PID object uses the Integral and Deriv(ative) numbers in terms of seconds, not minutes. • Gain – Proportional Gain is a multiplier that determines the ratio of process output response to the error. Gain can be a positive or negative number. A negative gain makes the PID "reverse acting." • Integral – Integral is used to modulate the process output to eliminate error. Integral is typically a small, positive decimal number. Integral should initially be set to zero (has no effect on the process output) until the Proportional Gain and Offset values have been determined. When adding Integral, add in increments of 0.01. • Deriv(ative) – Derivative causes the process output to be proportional to the rate of change of the process variable or error. Derivative is typically a small, positive decimal number. Derivative should initially be set to zero (has no effect on the process output) until the Proportional Gain and Offset values have been determined. When adding Derivative, add in increments of 0.001. • Formula – For future use. At this time, Formula must be kept at its default value of 1 (one). • Option – The Option component is used to prevent the phenomenon know as integral, or reset, windup. To minimize the effect of integral windup, you can restrict the integral component's maximum effect by defining an integral limit (Option). Option uses the same unit of measurement as the process output. When adding the Option component, start with the maximum expected or experienced steady state offset and adjust accordingly. Gain, Integral, Deriv(ative), and Option can be set either by using constant numbers (the letter C followed by a number, for example C1 for the value 1 or C.001 for the value 0.001; or by using an existing register (choose the desired register from the Register Selector).	

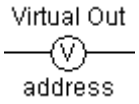
SELECTOR

Description	Example
Output the second input value if the first input is true; otherwise, output the third input value. Selector (and Move) are used to construct conditional statements. They test the first input to see if it is true. If it is true, one course of action is taken. If it is not true, another action is taken. Selector is an If-Then-Else statement and requires three inputs. The first input is the test, or conditional statement. The second input states what happens if the first input is true. The third input states what happens if the first input is not true. If the first input is true, then the second input's value becomes the output. else, if the first input is not true, then the third input's value becomes the output. Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.	

TOTAL

Description	Example
Sum the absolute value of a numeric input each time a new value arrives. This object has an optional reset input that enables you to reset Total at a specified time. If the reset is left blank, it is automatically reset at the start of each day (midnight). Total sums the absolute value of a numeric input each time a new value arrives. This object has an optional reset input that enables you to reset Total at a specified time. If the reset is left blank, HT3 will reset it at the start of each day (midnight). Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.	

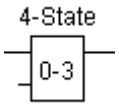
VIRTUAL OUT

Description	Example
Create a simulated output relay for reference elsewhere in ladder logic. Virtual Out objects must be assigned to an unused Register in the 30000 range. **NOTE:** An Analog Virtual Out is similar to a Digital Virtual Out except that the 0/Units and 1/Scale parameters are numeric values instead of On and Off.	

DIGITAL LADDER OBJECTS

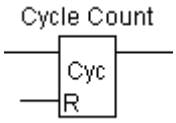
The following digital objects are provided in Logic Builder.

4-STATE

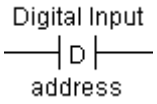
Description	Example
Combines two logical inputs to create a numeric output with a value from 0 to 3. 4-State fetches the state of two digital registers (inputs) and produces an output state, from 0 - 3, based on the conditions shown in the table below. Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.	

Value of First Input	Value of Second Input	Combined Output
0	0	0
1	0	1
0	1	2
1	1	3

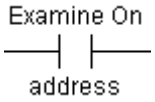
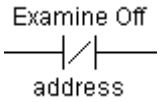
CYCLE

Description	Example
Cycle counts the number of times a logical input has turned on. This object features an optional reset input (a specific time to reset the timer). If left blank, the timer is automatically reset at midnight. Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.	

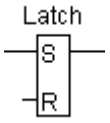
DIGITAL INPUT

Description	Example
Create a logical digital input that can be set to 0 or 1 and be used to control a register. Logic Builder features two Digital Input objects (Digital Input and Momentary Input) that are typically used to provide some type of operator control or input *into* the logic. This object can be controlled by a remote event through the use of an auto control configured in HT3. Optionally, you can enable direct operator control by including the input in a custom screen. (see the ***PLC033 Installation and Operation Manual*** for more information). The Control command can be used on this object to test your logic. Note that Digital Input objects remain at the value set through the Control command until they are manually reset from the Control command. Digital Input objects must be assigned to an unused Register in the 0-9999 range. **NOTE:** The Digital Input object does not use an input therefore it can't be used in AND functions unless it is the first object on the rung (see "Digital Input Example" on page 68)	

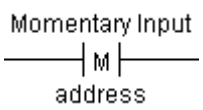
EXAMINE ON AND EXAMINE OFF

Description	Example
Logic Builder features two Digital Examine objects (Examine On and Examine Off) that fetch the current value of a real or logical register. When Logic Builder reaches an Examine object, it retrieves its current value and assigns that value to the Examine object. Examine On retrieves, or samples, the value of an existing Digital register and *"ands"* it with the input value. Examine Off is similar to Examine On, except that the retrieved, or sampled, value is inverted with a system-generated *"not"* equation. If the current value of the register is 0, Examine Off inverts the value so that it becomes 1. Examine objects must be provided with an *existing* Register. This can be the register assigned to a real physical input/output or the register number assigned to a logical output that was configured somewhere else in the current ladder logic diagram.	 

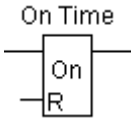
LATCH

Description	Example
Latch sets or resets the output when either input is true; otherwise, the previous value is kept. Latch is used to set and reset a digital output, (e.g., an auto control). It requires two inputs (conditions). The first input is used to set the conditions under which the register should be set. The second input sets the conditions under which the output should be reset. Each time the ladder logic diagram is scanned, these input conditions are evaluated. If either is true, the auto control's state changes from 0 to 1 (Set) or 1 to 0 (Reset). If neither condition is true, the auto control retains its previous value – state does not change. In the event that both conditions are true, the Set command overrides the Reset command. Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.	

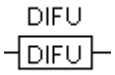
MOMENTARY INPUT

Description	Example
Momentary Input is a manual digital input that enables you to *temporarily* change its state (opposite of its initial value). A Constant or analog value that indicates the number of seconds that the Momentary Input is to stay on must be placed before Momentary Input. Logic Builder features two Digital Input objects (Digital Input and Momentary Input) that are typically used to provide some type of operator control or input *into* the logic. Direct operator control is enabled by including the input in a custom screen. The Control command can be used on this object to test your logic. Note that Digital Input objects remain at the value set through the Control command until they are manually reset from the Control command. Momentary Input objects must be assigned to an unused Register in the 0-9999 range. **NOTE:** The Momentary Input object uses a constant or analog value as an input therefore it can't be used in AND functions unless it is the first object on the rung (see "Momentary Input Example" on page 68).	

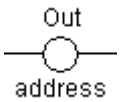
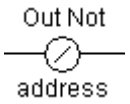
ON TIME

Description	Example
On Time calculates the number of seconds a logical input is in the On state. This object features an optional reset input (a specific time to reset the timer). If left blank, HT3 automatically resets the timer at midnight. Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.	

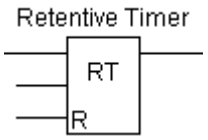
ONE-SHOT (DIFU)

Description	Example
DIFU = Differentiate Up Momentarily turn on an Output when the state of a specified Input changes from OFF to ON. The Output remains on (true) for the length of time it takes the system to complete one program scan. The One-Shot (DIFU) is reset only when the program scan is complete and the Input's state changes from On to Off. Any change in state during the program scan interval has no affect on the Output. One-Shot (DIFU) uses its Initial parameter to determine what result to produce. Setting Initial to 0 (default setting), results in a change from 0 to 1. Changing Initial to 1, results in a DIFD (Differentiate Down), i.e., the Output remains off (false) for one program scan. To use a One-Shot (DIFU) in a ladder logic diagram, place it directly after a digital Input. Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.	

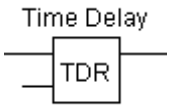
OUT AND OUT NOT

Description	Example
Logic Builder features two Digital outputs. These outputs are auto controls that output *from* the ladder logic diagram *to* a specified digital control register. Out Not is an auto control with the invert parameter set. Typically, the control register exactly follows the state of the register it is monitoring. Using Out Not, enables you to invert, or reverse, this relationship. Use Out to create an auto control that outputs from the diagram to a control register. Use Out Not to create an auto control with the invert parameter set. Out objects must be provided with an existing Register assigned to a real physical input/output.	 

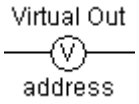
RETENTIVE TIMER

Description	Example
Similar to a Time Delay (see below), except that the timer isn't reset when the rung goes false. The value of the accumulated time is retained. When the rung goes true again , the object will begin accumulating time again starting at the retained value. This object has a third input that can be used to reset the timer. The reset should be a digital value (On to reset, Off to run). Use a Momentary Input to allow a user to manually reset the timer. If the third input isn't used, the timer will automatically reset at midnight. Retentive Timer uses its INITIAL parameter to determine what result to produce. Setting Initial to 0 (default setting), results in a change from 0 to 1. Changing Initial to 1, results in a Retentive Timer Off command. Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.	

TIME DELAY

Description	Example
Use a configurable timer to turn an output on (or off), provided the output's rung is true. When the timer expires, the output is turned on. If the rung goes false anytime before the timer expires, the timer is reset. A Constant is typically used to set the timer's maximum time, but any numeric value, for example an analog input (set point), could be used. Time Delay uses its INITIAL parameter to determine what result to produce. Setting Initial to 0 (default setting), results in a change from 0 to 1. Changing Initial to 1, results in a Time Delay Off command. Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.	

VIRTUAL OUT

Description	Example
This object can be used to create a simulated output relay for reference elsewhere in ladder logic. Virtual Out objects must be assigned to an unused Register in the 10000 range. **NOTE:** A Digital Virtual Out is similar to an Analog Virtual Out except that the 0/Units and 1/Scale parameters are On or Off instead of numeric values.	

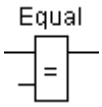
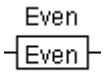
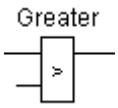
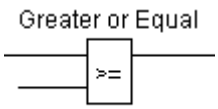
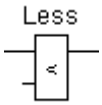
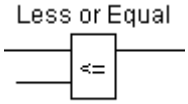
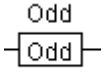
LADDER OPERATORS: DESCRIPTIONS AND PROPERTIES

COMPARE OPERATORS

Logic Builder's Compare operators function in much the same way that the Math operators do, but they output a 1 or 0 instead of a numeric value. All of the operators, with the exception of Even and Odd, require a second input.

Even and Odd evaluate a single input and output a 1 (one) if the statement is true. All of the other operators compare two inputs and output a 1 (one) if the comparison is true.

Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.

Description	Example
Equal – Output 1 if both inputs have the same value; otherwise, output 0.	
Even – Output 1 if the input's value is even; otherwise, output 0.	
Greater – Output 1 if the first input's value is greater than the second input's value; otherwise, output 0.	
Greater or Equal – Output 1 if the first input's value is greater than or equal to the second input's value; otherwise, output 0.	
Less – Output 1 if the first input's value is less than the second input's value; otherwise, output 0.	
Less or Equal – Output 1 if the first input's value is less than or equal to the second input's value; otherwise, output 0.	
Odd – Output 1 if the input's value is odd; otherwise, output 0.	

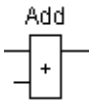
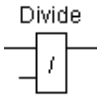
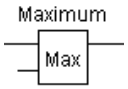
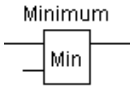
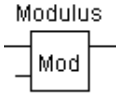
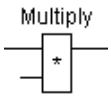
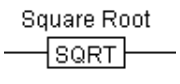
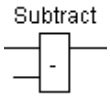
MATH OPERATORS

Each of Logic Builder's Math operators requires two inputs to perform its designated function. For example, Add fetches the first input's value and adds it to the second input's value; Divide fetches the first input's value and divides it by the second input's value; Minimum outputs the lesser value of the two inputs.

Math operators output numeric, or analog values in most cases. The exceptions are Divide, which outputs 0 if one or both of the inputs are zero, and Modulus, explained below, which outputs 0 if there is no remainder.

Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.

Modulus uses integer division to divide the first input by the second input and returns the remainder (i.e., the portion of the first input that isn't evenly divided by the second input). For example, dividing 15 by 6 returns a remainder of 3. Modulus outputs 0 if there is no remainder. For example, there would be no remainder when dividing 15 by 5, so Modulus would output 0.

Description	Example
Add – Output the sum of two inputs.	
Divide – Output the quotient that results when the first input's value is divided by the second input's value. Output 0 (zero) if the value of one or both of the inputs is 0.	
Maximum – Output the greater of two input values.	
Minimum – Output the lesser of two input values.	
Modulus – Output the modulus (remainder) that results when the first input's value is divided by the second input's value. Output 0 (zero) if there is no remainder.	
Multiply – Output the product that results when the first input's value is multiplied by the second input's value.	
Square Root – Output the square root of an input.	
Subtract – Output the difference between the first input's value and the second input's value.	

TIME / DATE OBJECTS

Logic Builder's Time and Date objects, when placed on a rung or branch, replace the rung's or branch's value with their own current value. Time and Date objects are useful when you need an event to occur at a certain time (e.g., you want to have an irrigation system come on and go off on certain days at specific times.) Time and Date objects do not require any inputs.

Registers for these objects are generated by Logic Builder when needed. It isn't necessary to manually assign them to a specific register in the Logical Memory Map.

Description	Example
Day of Month – Replace the value of the rung or branch with the day of the month (1-31).	—Day of Month—
Day of Week – Replace the value of the rung or branch with the day of the week (1-7, where Sunday = 1).	—Day of Week—
Day of Year – Replace the value of the rung or branch with the day of the year (1-366).	—Day of Year—
Hour – Replace the value of the rung or branch with the number of hours that have passed since midnight (0-23).	—Hour—
Minute – Replace the value of the RUNG or BRANCH with the number of minutes that have passed in the current hour (0-59).	—Minute—
Month of Year – Replace the value of the rung or branch with the month of the year (1-12).	—Month of Year—
Second – Replace the value of the rung or branch with the number of seconds that have passed in the current minute (0-59).	—Second—
Time of Day – Replace the value of the rung or branch with the number of seconds that have passed since midnight (0-86399).	—Time of Day—
Week of Year – Replace the value of the rung or branch with the week of the year (1-53).	—Week of Year—
Year – Replace the value of the rung or branch with the current year.	—Year—

PID BASICS

A Proportional Integral Derivative (PID) controller is a common feedback controller designed to drive a measurable process variable towards a desired value (a user-defined setpoint). The closed-loop controller compares the process variable's value to the setpoint. The difference between the two values is defined as the error:

Error = setpoint - process variable

The error between the setpoint and the process variable is used to calculate the process output value. The process output value controls a physical device that affects the process variable such that the process variable is driven towards the setpoint.

A PID controller looks at the error's current value, the amount and duration of the error (integral), and the rate of change of the error (derivative) to determine how much – and how long – corrective action should be applied to the process in order to bring the process variable's value back to the setpoint.

PID INPUTS

The PID object features three inputs:

- Process Variable (also called the process input) is the input you are trying to control (e.g., level, pressure, temperature, flow, or pH).
- Setpoint is the desired value of the process variable. For example, the setpoint for a car's cruise control would be 70 mph if you wanted to maintain a speed of 70 mph.
- Manual is optional. It is only necessary when the process variable has a manual switch that can be used to turn automatic control off and on. It is important to connect the manual switch to the PID object. When the two are connected and the process variable is in manual mode, the PID object is able to zero the integral part of the PID object to prevent accumulation of meaningless error values. The manual input can also be used to "zero" the accumulation of error when the process output is at its minimum and maximum value.

PID PARAMETERS

To set the PID's parameters, select the object and open Inspector.

The PID features the following parameters:

- **(Proportional) Gain** – Proportional Gain is a multiplier that determines the ratio of process output response to the error. Gain can be a positive or negative number. A negative gain makes the PID "reverse acting."
- **Integral** – Integral is used to modulate the process output to eliminate error. Integral gain sums all previous errors (integration) over time and multiplies this by the integral to affect the process output value. The result is that the PID output is proportional to the amount and duration of the error.

  Integral is typically a small, positive decimal number. Integral should initially be set to zero (has no effect on the process output) until the Proportional Gain and Offset values have been determined. When adding Integral, add in increments of 0.01.

- **Deriv(ative)** –Derivative causes the process output to be proportional to the rate of change of the process variable or error. Derivative action can compensate for a rapidly changing process variable. The Derivative part of the PID algorithm evaluates the process variable's rate of change in order to prevent any oscillation caused by integration. This component is proportional to the rate of change of the process variable. Derivative causes the process output to decrease if the process variable is increasing rapidly.

  Derivative is typically a small, positive decimal number. Derivative should initially be set to zero (has no effect on the process output) until the Proportional Gain and Offset values have been determined. When adding Derivative, add in increments of 0.001.

- **Formula** – For future use. At this time, Formula must be kept at its default value of 1 (one).
- **Option** –The Option component is used to prevent the phenomenon known as integral, or reset, windup. When a sustained error occurs, the integral term can become quite large. This eventually causes the process output to become saturated. Integral windup results when the integral term continues to build up while the controller is saturated. To minimize the effect of integral windup, you can include an integral limit (Option). Option defines the maximum contribution of the equation's integral component; the integral component cannot effect the output of the PID object more than the value of Option.

  Note that ***Option*** uses the same unit of measurement as the output of the PID object. When adding the Option component, start with the maximum expected or experienced steady state offset and adjust accordingly. If Option is set too low, the output of the PID object will continue to experience steady state offset. If Option is too high, the output of the PID object will encounter windup and more overshoot.

Gain, Integral, Deriv(ative), and Option can be set using constant numbers (the letter C followed by a number, for example C1 for the value 1 or C.001 for the value 0.001); or an existing register (choose the desired register from the Register Selector).

STRATEGY FOR IMPLEMENTING PID CONTROL

The output of the PID object is directly calculated by this equation:

$$Process\ Output = Kp(Error + Ki\int(error)\Delta t + Kd(\Delta Error / \Delta t))$$

Where,

Kp = proportional gain
$Error$ = setpoint - process variable
Ki = integral gain
Δt = change in time
Kd = derivative gain
$\Delta Error$ = change in error

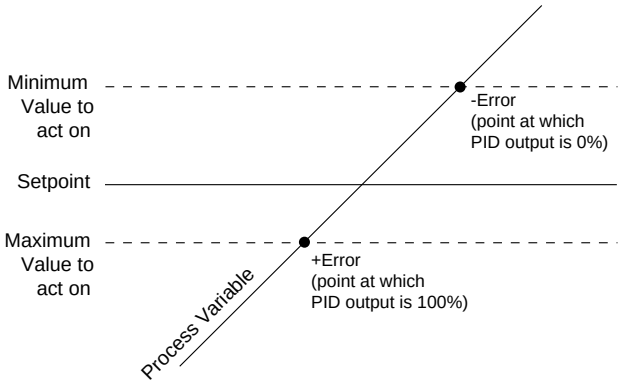
In the equation above, the error, integral, and derivative are added together and then multiplied by a tuning constant (proportional gain) to produce the process output.

IMPORTANT: The algorithm used by Logic Builder's PID object calculates the Integral and Deriv(ative) numbers in terms of *seconds*, not minutes.

DETERMINING PROPORTIONAL GAIN

The first step is to determine the range of error around the setpoint (maximum and minimum values of error to be acted on). These values are determined by the system engineer's requirements. Remember that Error = Setpoint - Process Variable.

For a "normal" PID, the maximum error is the point at which the PID output is at 100%; the minimum error is the point at which the PID output is at 0%. For a reverse-acting PID, these relationships are reversed.



The next step is to determine the valid output range – in engineering and raw units – of the process output (the maximum and minimum acceptable values). For example, a 4-20mA device using DFS protocol could have a range of 820-4095; a Modbus-compatible device whose acceptable output range is 0-100% could have a range of 0-32767.

Keep in mind that the PID output, the process output's valid range, and the offset (discussed below) must all be in the same units as the process output. If the process output is measured in percent (%), all of these variables must also be measured in percent (%). This means that not only is the scale of the units being changed in the PID object, but the units themselves must also be changed.

Proportional Gain is derived using the values for the process output's valid range and the values of the +Error and -Error.

Use the following equation to derive Proportional Gain:

$$Kp = (max\ out - min\ out) \div (|+Error| + |-Error|)$$

Where,

Kp = Proportional Gain

max out = maximum value of process output's valid output range

min out = minimum value of process output's valid output range

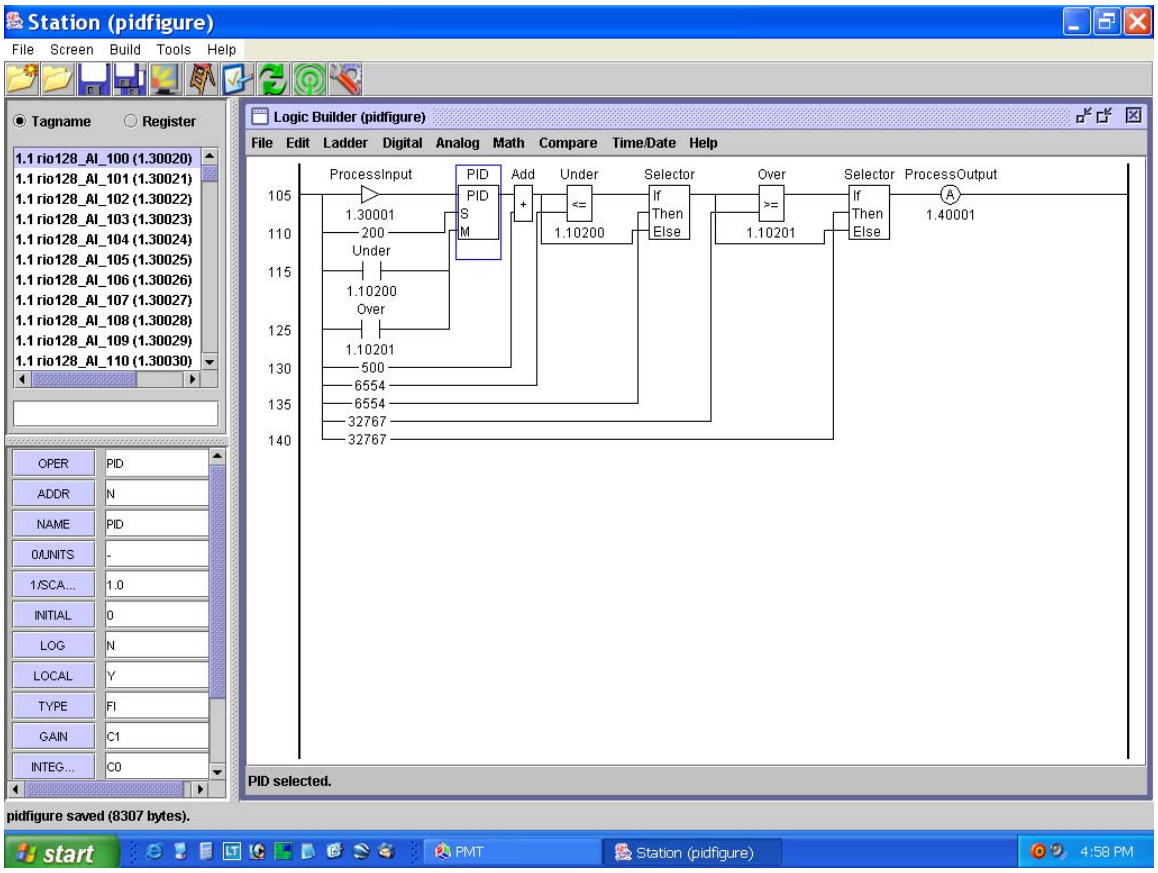
|+*Error*| = absolute value of the process variable's +Error (setpoint - process variable's maximum error)

|-*Error*| = absolute value of the process variable's -Error (setpoint - process variable's minimum error)

DETERMINING OFFSET

Offset's purpose is to put the PID output in the correct range. Offset is the typical/average/mean output of the process output. Offset is added *directly* to the PID output as shown in the ladder below.

SAMPLE LADDER



Notes

SCREEN BUILDER OVERVIEW

Screen Builder is an application included in the Process Management Toolkit that lets you create a graphical representation of your system. By building a screen – using text, images, objects, and animation – and then linking the screen's components to actual I/O, you can get a quick, real-time view of your operation. These screens aren't limited to showing status, they can also be used to control hardware.

For example, you can build a screen that shows the level in a well and the run status of pumps, and then link it to the physical hardware located in the field using the hardware's corresponding register. This linking lets you create a virtual picture of how the equipment is operating; the screen mimics the activity of the equipment. If the screen showed the well reaching a critical level, you could turn on a pump by simply clicking a button on the screen.

NOTE: Screens can be downloaded from the Hyper Server Module. See Retrieving a Screen from an HSM beginning on page 111 for more information.

The image below is an example of the type of screen you can create for your system.

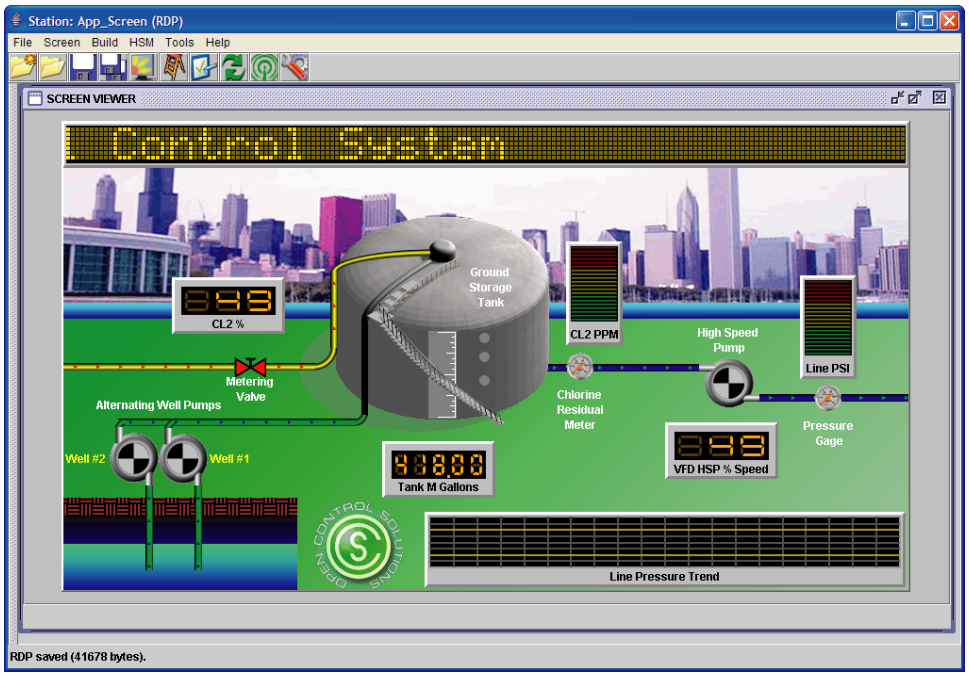


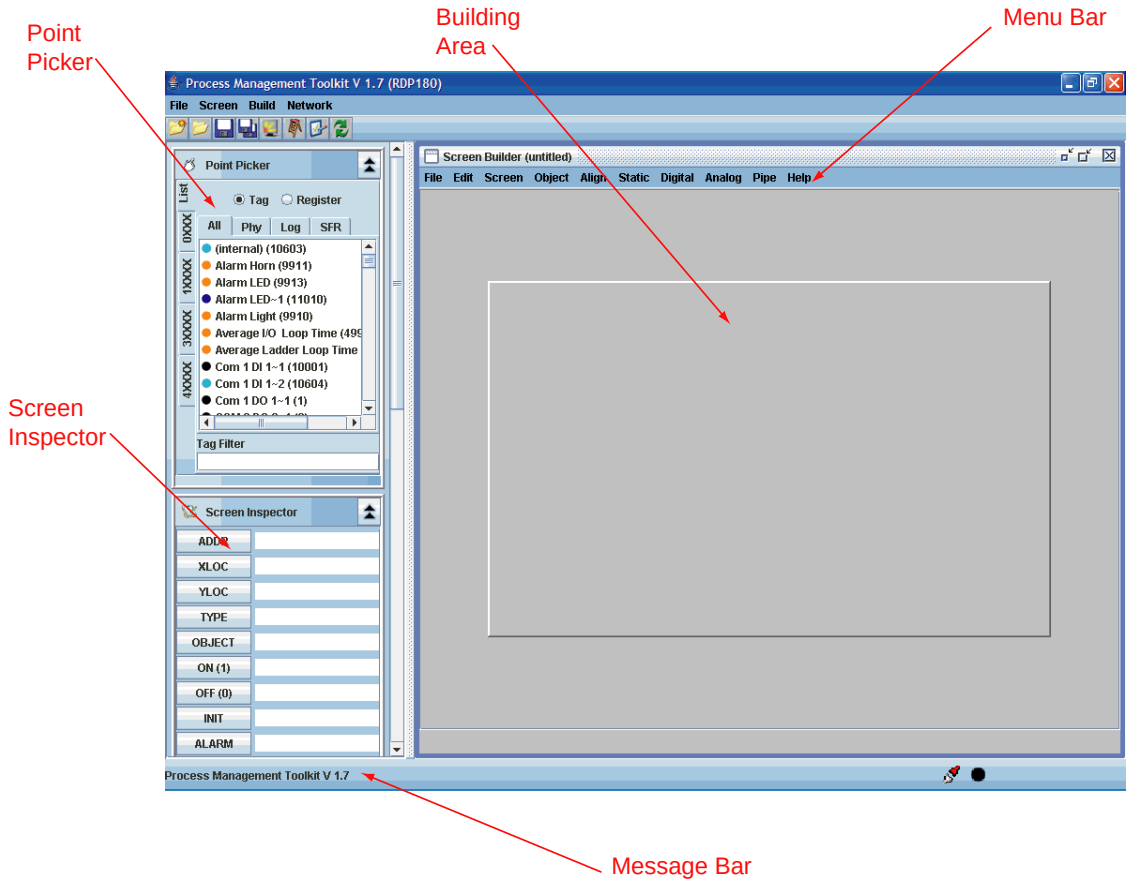
Figure 5-1, Custom Screen Example

OPENING SCREEN BUILDER

To open Screen Builder, select Screens from the Build menu. Screen Builder opens in the right panel, and Screen Inspector [see “Editing Properties: Logic Inspector and Point Picker” on page 81] is added to the list of tools in the left panel.

SCREEN BUILDER INTERFACE

Before you begin using Screen Builder, it is helpful to familiarize yourself with the application's graphical user interface (GUI). The sections below describe the components that make up Screen Builder's GUI – its menus and menu commands, the building area, the message bar, the register selector, and Inspector.



SCREEN BUILDER GUI

The basic components of Screen Builder's GUI (shown on the previous page) are:

- **Menu Bar** – Features ten menus with commands. These commands are used to build, test, and save your screens.
- **Building Area** – Area where you build your custom screens. The first time you open Screen Builder, a blank screen is displayed.
- **Message Bar** – Displays messages, such as prompting messages and error messages, that provide you with additional information and instructions on Screen Builder functions.

Additionally, Point Picker and Screen Inspector appear in PMT's tool panel.

- **Point Picker** – Displays a list of available registers. The list can be sorted by tag name, register, or I/O type. See "Point Picker Basics" on page 21 for more information.
- **Screen Inspector** – Displays the selected object's parameters.

MENUS

Note that the menus below are organized according to how they appear (from left to right) in Screen Builder.

FILE MENU

From the File menu, you can perform the following actions:

- **New** – Create a new blank screen.
- **Load** – Load (open) a screen file from the current project.
- **Name Screen** – Give the current screen a unique name and save it to the current project.
- **Copy Screen As** – Copy the current screen and give it a new, unique file name.
- **Import Screen** – Import a .CSV screen file. If the imported screen uses any custom images (images that are not part of the default image library), they must be manually copied to the images folder. The images folder is a subdirectory of the main PMT directory (usually C:\Program Files\PMT2\Images).
- **Import HSM Screen** – Import a screen from an HT3 system (Hyper Server Module). You will be prompted to enter either the host name or IP address of the HSM, and then you will be prompted to enter your HT3 login name and password.
- **Export Screen** – Export the current screen to a .CSV file. This allows you to copy the screen to another computer or import it into another project. Note that this does not export the images – only the screen framework. If the screen is going to be imported on another computer, any custom images (images that are not part of the default image library) must be manually copied to the images folder in the other computer. The images folder is a subdirectory of the main PMT directory (usually C:\Program Files\PMT2\Images).
- **Remove** – Remove a screen from the current project.
- **Close** – Close Screen Builder.

EDIT MENU

From the Edit menu, you can perform the following actions:

- **Undo** – Undo the last action.
- **Cut** – Remove the selected item and place it on the clipboard.
- **Copy** – Copy the selected item and place it on the clipboard.
- **Paste** – Paste text from the clipboard to a new location.
- **Delete** – Permanently remove the selected item.

SCREEN MENU

From the Screen menu, you can perform the following actions:

- **Refresh** – Update the status of all objects on the screen.
- **Station** – Change the station number for all objects on the screen. The addresses of all objects that include the old station number will be changed to reflect the new station number.
- **Size** – Set the size of the screen (in pixels). The default screen size is 600 x 420 pixels. Maximum screen size is 1024x768 or your current monitor resolution (whichever is less).

OBJECT MENU

From the Object menu, you can perform the following actions:

- **Refresh** – Update the status of the selected object. If the object is static, the Message bar displays “Cannot refresh STATIC object.”
- **to Front** – Move the selected object to the very front (top) when objects are layered* (stacked) one on top of the other.
- **Forward** – Move the selected object forward one layer when objects are layered* (stacked) one on top of the other.
- **Backward** – Move the selected object backward one layer when objects are layered* (stacked) one on top of the other.
- **To Back** – Move the selected object to the very back (bottom) when objects are layered* (stacked) one on top of the other.
- **Duplicate** – Create an exact duplicate of the selected object or group of objects. See "Selecting Multiple Objects" for instructions on creating object groupings.
- **Select Behind** – Select the object directly behind the currently selected object. This function is useful when an object is "hidden" by another object.

* Layering refers to *when* an object was added to the screen. Objects are layered based on the sequence in which they are added. The first object is placed at the bottom of the stack and subsequent objects are positioned above it in the order in which they are added.

ALIGN MENU

From the Align menu, you can perform the following actions:

- **Left** – Align the selected object(s) horizontally along the left edge of the reference object (the first object selected, or clicked).
- **Center** – Align the selected object(s) horizontally in the center of the reference object.

- **Right** – Align the selected object(s) horizontally along the right edge of the reference object.
- **Top** – Align the selected object(s) vertically along the top edge of the reference object.
- **Middle** – Align the selected object(s) vertically in the middle of the reference object.
- **Bottom** – Align the selected object(s) vertically along the bottom edge of the reference object.

STATIC MENU

From this menu, you can add static objects to your screen. Static objects are those that have no status – they are not linked to digital, analog, or virtual points. They are usually added for visual effect. Descriptions and examples of these objects, along with a list of their properties, can be found in the section titled Static Screen Objects beginning on page 167.

Most of these items are scalable. Scalable means that an object's properties – its size, text color, font size, etc. can be changed. Objects that are image files (.gif) are not scalable. For example, you cannot increase the size or color of a spinner or a graphic. Other objects such as rectangles and 3D rectangles are scalable. A rectangle's size and border can be increased. You can change the color and size of its text.

From this menu, you can add any of the following objects:

- Text
- 3D Text
- Banner Text
- Image (not scalable)
- Rectangle
- 3D Rectangle
- Round Rectangle
- Oval
- Tick Mark
- Grid
- Gradient
- Pattern: checker, brick, sand, grass, diamond, steel, rain, or earth

DIGITAL MENU

From this menu, you can add digital objects to your screen. Digital objects are linked to digital-type registers. Descriptions and examples of these objects, along with a list of their properties, can be found in the section titled “Digital Screen Objects” beginning on page 141.

Note that Control Rectangle and Graphic Control are the only objects that can be used to *control* a digital point.

Many of these items are scalable. Scalable means that an object's properties – its size, text color, font size, etc. – can be changed. Objects that are image files (.gif) are not scalable. For example, you cannot increase the size or color of a spinner or a graphic. Other objects such as rectangles and 3D rectangles are scalable. A rectangle's size and border can be increased. You can change the color and size of its text.

From this menu, you can add any of the following objects:

- Text
- Graphic (not scalable)
- Rectangle
- 3D Rectangle
- Round Rectangle
- Oval
- Arrow
- Animation (not scalable)
- Switch
- Graphic Control (not scalable)
- Control Rectangle
- 4-State Rectangle
- 4-State Graphic (not scalable)
- 4-State Text

ANALOG MENU

From this menu, you can add analog objects to your screen. Analog objects are linked to analog-type registers. Descriptions and examples of these objects, along with a list of their properties, can be found in the section titled “Analog Screen Objects” beginning on page 123

Most of these items are scalable. Scalable means that an object's properties – its size, text color, font size, etc. – can be changed. Objects that are image files (.gif) are not scalable. For example, you cannot increase the size or color of a spinner or a graphic. Other objects such as rectangles and 3D rectangles are scalable. A rectangle's size and border can be increased. You can change the color and size of its text.

From this menu, you can add any of the following objects:

- Text
- LED Text
- Panel (not scalable)
- Gauge (not scalable)
- LED Gauge
- Dial
- Rotary
- Bar Graph
- LED Bar
- Color
- Control
- Slider
- Trend
- Time
- Time Control

PIPE MENU

From this menu, you can add pipe and elbow objects to your screen. Some of these objects can be linked to analog or digital registers; others are static. Descriptions and examples of these objects, along with a list of their properties, can be found in the section titled Pipe Screen Objects beginning on page 159.

All of these items are scalable. Scalable means that an object's properties – its size, text color, font size, etc. – can be changed. Objects that are image files (.gif) are not scalable. For example, you can change the background color and width and height of these objects.

From this menu, you can add any of the following objects:

- Gradient Pipe
- Gradient Elbow
- Valve
- Spinner
- Digital Pipe
- Digital Elbow
- Static Pipe
- Static Elbow

HELP MENU

- About – Display Screen Builder’s version number and release date in the Message bar.

WORKING WITH SCREENS

CREATING A NEW SCREEN

1. Choose **New** from the **File** menu. A new screen is created.
2. If a screen with unsaved changes is open and you attempt to create a new screen, a dialog box appears with a message telling you that the current screen hasn't been saved and asking if you want to discard these changes.
3. Click **Cancel** to exit the dialog box. Save your work and then try creating a new screen again. If you don't want to save the changes made to the current screen, click **Discard**. Any changes made to the current screen are *not* saved and a new screen is created.

SETTING THE SCREEN'S SIZE

Once you have created a new screen, you can begin setting its specifications, starting with screen size. With the Size command, you can set the dimensions for your screen.

The default size is 600x420 (measured in pixels), but you can increase and/or decrease the height and width. The maximum screen size is 1024x768 or your current monitor resolution (whichever is less).

To avoid horizontal and vertical scrolling when viewing a screen:

- Width = maximum width – 100 pixels (or 300 pixels if you want to be able to view the Inspector and point picker information)
- Height = maximum height – 200 pixels.

To set the screen's size:

1. Choose **Size** from the **Screen** menu.
2. Enter the desired width and height.
3. Click **Resize**.

SAVING A SCREEN

Screen Builder features three methods for saving screens. All of these can be found in the **File** menu:

- **Name Screen** – Use this option the first time you save your screen. When you name a screen, you make it part of the current PMT project. However, the screen is *not* transferred to the PLC. The screen's definition, or framework, is transferred to the PLC when the project's configurations are installed (see Install Configurations on the PLC on page 61). Be aware that *only* the screen's definition is transferred. Images are stored locally on the computer that was used to create the project. If the screen is going to be viewed on another computer, any custom images (images that are not part of the default image library) must be manually copied to the images folder on the other computer. The images folder is a subdirectory of the main PMT directory (usually C:\Program Files\PMT2\Images).
- **Copy Screen As** – This option works similar to **Name Screen** except that it gives you the opportunity to give the screen a different name. This is useful if you want to use the current screen as a template for another screen. Once you've copied the screen and given it a new name, you can make any desired changes.
- **Export** – Use this option if you want to make the screen "portable." Export saves the screen's definition, or framework, as a .CSV file. This .CSV file can then be copied to another computer or imported into a new project. If the screen is going to be imported on another computer, any custom images (images that are not part of the default image library) must be manually copied to the images folder on the other computer. The images folder is a subdirectory of the main PMT directory (usually C:\Program Files\PMT2\Images).

OPENING AN EXISTING SCREEN

Screen Builder features two methods for opening screens. Both of these can be found in the File menu:

- **Load** – Use this option to open a screen that is part of the current project. Load can be used to open files that were saved using either the **Name Screen** or **Copy Screen As** commands.
- **Import** – Use this option to import a .CSV screen file. If the screen was created on another computer and used custom images (images that are not part of the default image library), you must copy the custom images into the images folder on your computer. The images folder is a subdirectory of the main PMT directory (usually C:\Program Files\PMT2\Images).

REMOVING A CUSTOM SCREEN

Screen Builder's **Remove** command, which can be found in the **File** menu, provides you with the option of removing obsolete screens from a project.

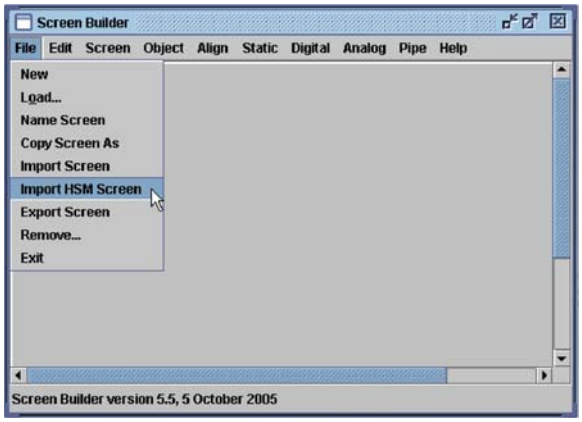
To remove a screen, choose **Remove** from the **File** menu, select the name of the screen to be removed and click **Remove**. At the **Are you sure?** dialog box, click **Remove** again.

IMPORTANT: When you remove a screen, it is permanently deleted from the project. It cannot be restored unless it was exported as a .CSV file. Any screens that include links to the removed screen need to be updated.

RETRIEVING A SCREEN FROM AN HSM

Retrieving screens requires that the computer running the PMT software be able to connect to the HSM that contains the desired screens.

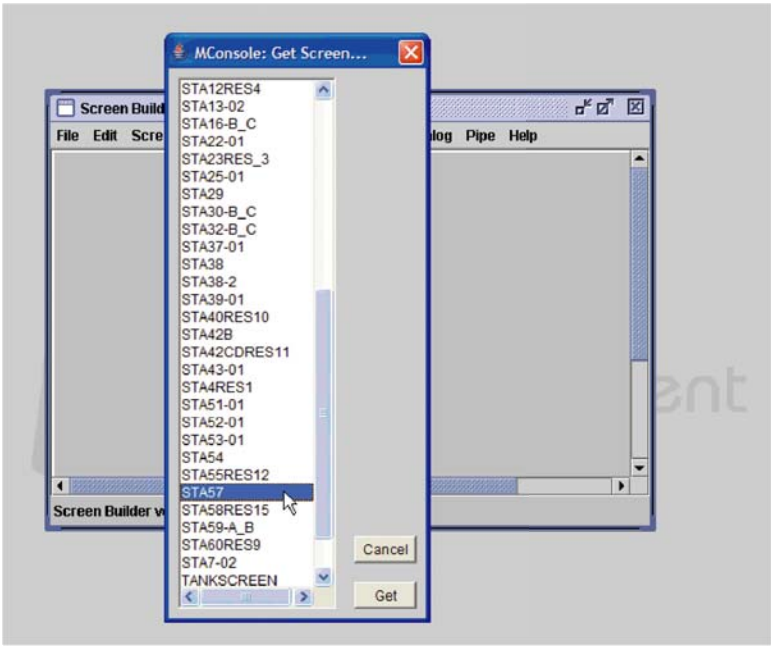
1. Launch PMT. Open the project file that you want to import the screen into.
2. Open Screen Builder.
3. Select **Import HSM Screen** from Screen Builder's **File** menu.



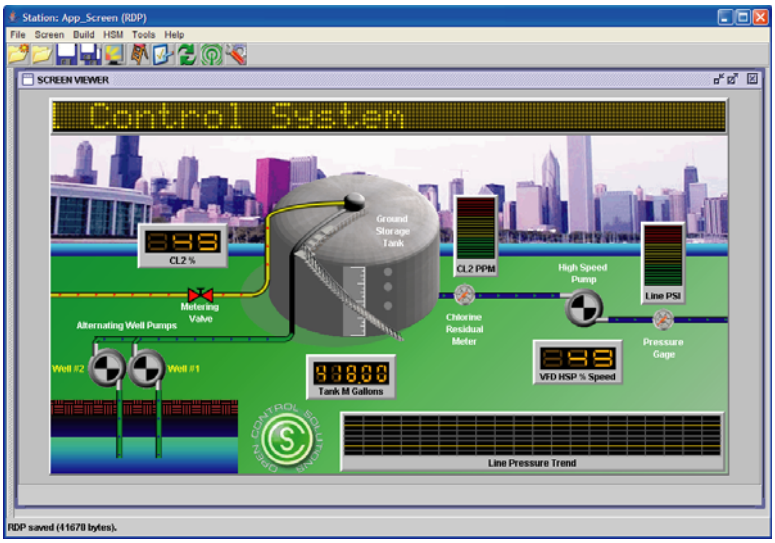
4. Enter the HSM's IP address or host name and click **OK**.

5. Enter your Login Name and Password and click **Login**.

6. Select the name of the screen you want to import and click **Get**.



7. The screen is imported and opens in Screen Builder. Save your project.



Refreshing and Animating Screens

With the **Refresh** command, you can update the status of all the objects on your screen or you can choose one object to refresh. It's similar to taking a static picture of your screen showing the current status and activity.

The Animate command allows you to see how the animated objects on your screen will look and behave when receiving live data.

Refresh the Status of a Single Object

1. Select the object you want to refresh by clicking it.
2. Choose **Refresh** from the **Object** menu. The object's current status is visible on the screen.

Refresh the Status of the Entire Screen

1. Choose **Refresh** from the **Screen** menu. The current status of all active objects is visible on the screen.
2. As you are building your screen, periodically refresh it to see how it will look in action.

Animate an Object

1. Select the object you want to animate by clicking it.
2. If you have entered a live telemetry address for the object, choose **Refresh** from the **Screen** menu to update the object's status. If you want to test the animation without entering a live telemetry address, enter a value in Inspector's Value box.
3. Choose **Animate** from the **Screen** menu.
4. To turn off animation, click anywhere on your screen.

CHANGING STATION NUMBER REFERENCES

The Station command enables you to find a specific station number referenced in the currently opened screen and replace it with a new station number prefix.

1. Open the screen to edit. Choose **Copy Screen As** from the **File** menu to save the screen with a new file name
2. Choose **Station** from the **Screen** menu. The **Screen Builder: Station** dialog box appears.
3. Enter the old prefix in the **Old Prefix** box and the new prefix in the **New Prefix** box.
4. Click **Change** to change the station number prefix or **Cancel** to exit without making changes. The Message bar reads "[#] address fields replaced" (where # indicates the number of object addresses changed).

WORKING WITH OBJECTS

This section explains how to add and manipulate objects. It covers topics including moving objects, working with layers, and editing the properties of objects. For information on specific screen objects, including descriptions, properties, and examples of how the objects look, see “Screen Objects: Descriptions and Properties” beginning on page 122.

ADDING AN OBJECT

Choose an object from one of the object menus (Analog, Digital, Pipe, or Static). The object appears on the screen and the Inspector window displays the object’s properties.

You are now ready to begin manipulating the object's properties. (Note you must press the ENTER key each time you make a direct change to any input box in the Inspector window. If you enter incorrect data in any Inspector input box, the message "***THAT ENTRY IS NOT VALID***" appears at the bottom of the Inspector window.)

DEFINING AN OBJECT: SCREEN INSPECTOR AND POINT PICKER

SCREEN INSPECTOR

Screen Inspector can be found on the left side of the main PMT window. It opens and remains docked in the tools panel whenever Screen Builder is opened.

By default, Screen Inspector is placed at the bottom of the tool panel. However, you can move it anywhere in the list of tools by clicking the Screen Inspector title (the cursor changes from an arrow to a box – see image below) and dragging Screen Inspector up or down to place it in a new location.

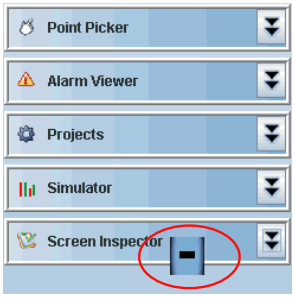


Figure 5-2, Moving Screen Inspector

When an object is added to a screen, Screen Inspector displays that object's default properties. The example shown below is for a digital text object. The fields displayed in Screen Inspector vary depending on the object type. See "Screen Objects: Descriptions and Properties" for detailed information on the fields listed in Screen Inspector for each object.

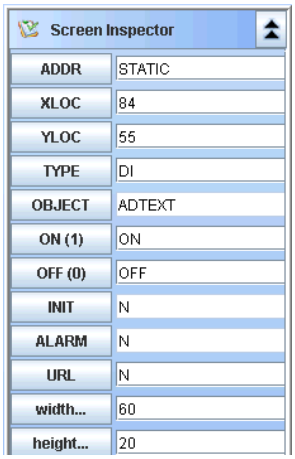


Figure 5-3, Screen Inspector

For most fields, you simply type the desired information into the box and press Enter. If you click the URL label or a label that requires you to select a color (e.g., textColor, backgroundColor) or an image, a

dialog box appears that enables you to choose the desired color, image or destination screen to link to. (See "Choosing Colors for Text and Objects," and "Choosing Images" for more information.)

POINT PICKER

Point Picker, located in the left panel of PMT, can be used to choose the register that you want to associate with an object.

1. Select the object (blue box appears around object).
2. Locate the register in Point Picker and double-click it. The ADDR field in Logic Inspector is updated to show the selected register; the tag name associated with the register will appear in the NAME field.

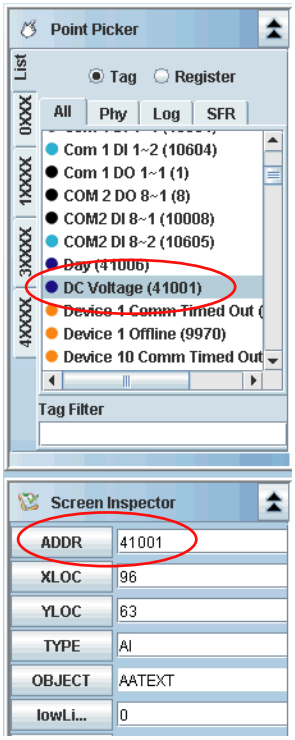


Figure 5-4, Selecting a Register for a Screen Object

CHOOSING COLORS FOR TEXT AND OBJECTS

Some of Screen Builder's objects require that you specify the color of the object's text, or its background or border.

1. Select the desired object.
2. In Inspector, click the name of the field whose color you want to set. The **Color** dialog box appears. The color sample box and the RGB boxes show the default color.

3. Choose a color from the list (its hexadecimal RGB number appears in the RGB boxes) or type an RGB number in the three RGB boxes. The color appears in the color sample box. To brighten or darken the color, select **Bright** or **Dark**.
4. Click **OK** to accept your changes. To exit the **Color** dialog box without accepting changes, click **Cancel**.

CHOOSING IMAGES

The following Screen Builder objects allow you to select their associated image, or graphic:

- Static Image
- Digital Graphic
- Digital Animation
- Digital Graphic Control
- Digital 4-State Graphic
- Analog Bar Graph
- Analog Control

All of the image files are stored in a directory named "images," which is a subdirectory of the main PMT directory (usually C:\Program Files\PMT2\Images). If you want to use images of your own in your custom screens (for example, maps or photos of equipment or buildings), place the files in the images directory. The images *must* be .gif files.

Initially, when you add one of these objects to a custom screen, their default image appears. To choose an alternate image:

1. Select the Screen Builder object whose image you want to change. The selected object will be surrounded by thin black lines.
2. In Inspector, click the name of the field that contains the image. For example, for a Static Image, you would click **image**.

NOTE: The Digital objects listed above allow you to select a different image for each possible state. For example, for the Digital Graphic, you can select a separate image for the on, off, and initial states by clicking the ON (1), OFF (0), and INIT field names. Detailed information on these objects can be found in "Screen Objects: Descriptions and Properties" on page 122.

3. When you click the field name, the **Image** dialog box appears. This dialog box lists all of the available images by their file name. If you want to see a thumbnail preview (a reduced-size version) of the image before placing it on your screen, select the **Preview** option and click the file name of the image.
4. When you decide on an image, click its file name to select it and then click **OK**. The selected image appears on your screen.

LINKING TO ANOTHER CUSTOM SCREEN

Create a link between an object (source) and another custom screen (destination)

1. Select the source object.
2. In Inspector, click **URL**. The **Link** dialog box appears.
3. Locate the destination screen, select it and click **Link**. The URL for the destination screen appears in the **URL** box. To exit the **Link** dialog box without creating a link, click **Cancel**.

Edit a link:

1. Select the object that contains the link you want to edit.
2. In Inspector, click **URL**. The **Link** dialog box appears.
3. Locate the new destination screen, select it and click **Link**. The URL for the new destination screen appears in the **URL** box. To exit the **Link** dialog box without making changes, click **Cancel**.
4. Save the screen by selecting **Save** from the main console's **File** menu.

MOVING AND MANIPULATING OBJECTS

Layering refers to *when* an object was added to the screen. Objects are layered based on the sequence they are added. The first object is placed at the bottom and subsequent objects are positioned above it in the order in which they are added.

SELECTING "HIDDEN" OBJECTS

The **Select Behind** command is useful when you have a "stack" of layered objects and you're trying to select one that is "hidden" in the stack. It is located somewhere in the middle or at the bottom of the stack, and there isn't any way to simply click on it to select it.

1. Select the object that is on the top of the stack.
2. Choose **Select Behind** from the **Object** menu (or hold down the CTRL key and click the object again). This selects the object directly beneath the top object. The Message bar tells you what type of object is selected.
3. Continue choosing **Select Behind** (or holding down the CTRL key and clicking the top object) until you have the desired object selected. The Message bar tells you what type of object is selected. View the object's properties in Inspector to be sure the correct object is selected.

NOTE: If you are using Windows, you can use either the CTRL or ALT key to select a hidden object.

SELECTING MULTIPLE OBJECTS

Grouping objects is useful when you create an arrangement of objects and you want to perform the same function on them simultaneously. For example, you may want to copy that arrangement to another screen or duplicate the arrangement within the current screen. This can be accomplished using the SHIFT key and selecting (clicking) each object, or by drawing a box around the desired objects.

Using SHIFT

1. Select an object in the group by clicking it.
2. Hold down the SHIFT key and then select the other objects in the group. As you do this a black line appears around the selected objects. The Message bar displays "[#] objects selected," where # is equal to the number of objects that are selected.
3. With the black line still visible, choose the editing command you want to perform, e.g., **Copy**, **Duplicate**, etc.

Drawing a Box Around Objects

1. Position your mouse pointer somewhere outside the area where the objects are located. You can use the area outside the screen boundaries if necessary. Hold down the left mouse button. The pointer becomes a cross hair.
2. Continue holding down the left mouse button and start dragging the cross hair to form a box around the objects.
3. When all the desired objects are within the box, release the mouse button. All the objects within the box are selected and the Message bar displays "[#] objects selected," where # is equal to the number of objects that are selected.
4. With the box still visible, choose the editing command you want to perform, e.g., **Copy**, **Duplicate**, etc.
5. If you need to start your box with the mouse pointer over an object, such as a background picture, do the following:
   A. Click in the area outside the screen boundary to deselect all objects. Check the Message bar to verify that no objects are selected.
   B. Hold down the SHIFT key as you press the left mouse button to start your box. Continue holding down the left mouse button and the SHIFT key and start dragging the cross hair to form a box around the objects

MANIPULATING LAYERED OBJECTS

Screen Builder features four commands that aid you in the process of manipulating and moving layered objects. They are:

- **to Front** (moves the object to the very top of the "stack")
- **Forward** (moves the object up one layer)
- **Backward** (moves the object down one layer)
- **to Back** (moves the object to the very bottom of the "stack").

As your screens become more complex, there will be times when they have multiple layers of objects. Layering objects gives your screens more depth and detail. But what do you do when you have three objects layered and you want to move the middle layer to the top of the "stack"? And you want to do this without "unlayering" the objects.

1. Select the object that you want to move by clicking it.
2. Choose one of the commands listed above (to Front, Forward, Backward, to Back).

DUPLICATING AN OBJECT

The Duplicate command is useful when you want to make an exact copy, or clone, of an object or group of objects. For example, instead of adding a new object and then editing its fields to match those of another object, you simply duplicate it.

1. Select the object or group of objects you want to duplicate by clicking it. See "Selecting Multiple Objects" for instructions on grouping objects.
2. Choose **Duplicate** from the **Object** menu. An exact copy of the original object or group of objects appears on the screen.

ALIGNING OBJECTS

The Align menu is a tool that helps create uniformity and add symmetry to your screens.

You may want to align two or more objects so that they are centered exactly on top of one another. Or you may have a row of objects that you want to align so that they are positioned along the same invisible horizontal line.

Objects can be aligned horizontally (left, center, right) and vertically (top, middle, bottom).

1. Select a reference object by clicking it. This is the object to which the others will be aligned.
2. Hold down the SHIFT key and then select the secondary objects. As you do this a black line appears around the selected objects. At the bottom of the Screen Builder window, you will see a message telling you how many objects have been selected.
3. Choose how you want the objects to be aligned (**Left**, **Center**, **Right**, **Top**, **Middle**, or **Bottom**). You can align the objects horizontally and then vertically by keeping them selected (make sure they are still surrounded by the black line) and then selecting another alignment direction from the **Align** menu.

MOVING AND POSITIONING OBJECTS

There are three ways to move and position your screen objects.

- Select the object to be moved, continue holding down the left mouse button (the pointer becomes a four-headed arrow) and drag the object to a new location.
- Select the object to be moved, and while it is still selected, use the arrow keys on your keyboard to move the object one pixel at a time.
- Move the object to a specific location by changing the XLOC (horizontal position) and the YLOC (vertical position) values in Inspector. A 0 (zero) in both of these boxes places the object in the top left corner of the screen. Increasing the XLOC value moves the object farther to the right. Increasing the YLOC value moves it farther down the screen.

MOVING MULTIPLE OBJECTS SIMULTANEOUSLY

It is possible to move a group of objects to a new position. You can keep the objects' relationship to one another the same as you move them. The process is similar to the one used when aligning objects.

1. Select the objects to be moved using one of the techniques described in "Selecting Multiple Objects," above.
2. Move the group using one of the procedures below:

- For finer control of movement, use the arrow keys on your keyboard to move the selected objects one pixel at a time.
- Click the selected objects. The pointer becomes a four-headed arrow. Hold down the left mouse button and drag the objects to the new location.

SIZING OBJECTS

There are two methods for setting an object's size:

- Click the object. Handles (small white boxes) appear around the object. "Grab" a handle by clicking it. Notice that all the other handles become black in color. Hold down the left mouse button and drag the selected handle to increase or decrease the object's size.
- Set the object to a specific size by entering values (measured in pixels) in the widthDim and heightDim boxes in Inspector.

USING THE EDIT MENU

The edit menu features standard commands for:

- Reversing the last action you performed (Undo)
- Cutting, copying, and pasting (Cut, Copy, Paste)
- Deleting objects from your screen (Delete)

Undo

When you perform an action on an object – you change the text color, move the object, etc. – and you want to reverse that action, choose **Undo** from the **Edit** menu.

Cut and Paste

To remove an entry from a data field and paste it in another location:

1. Highlight the data to cut and press CTRL + X.
2. Place your cursor in the destination field and press CTRL + V.
3. To remove an object or group of objects and paste in another location:
4. Select the object or group of objects and choose **Cut** from the **Edit** menu. See "Selecting Multiple Objects" for instructions on grouping objects.
5. Choose **Paste** from the **Edit** menu.

To remove an object or group of objects and paste in another screen:

1. Select the object or group of objects and choose **Cut** from the **Edit** menu. See "Selecting Multiple Objects" for instructions on grouping objects.
2. Open the screen to which to paste the object or objects.
3. Choose **Paste** from the **Edit** menu.

Copy and Paste

To copy an entry from a data field and paste it in another location:

1. Highlight the data to copy and press CTRL +C.
2. Place your cursor in the destination field and press CTRL + V.

To copy an object or group of objects and paste in another location:

1. Select the object or group of objects and choose **Copy** from the **Edit** menu. See "Selecting Multiple Objects" for instructions on grouping objects.
2. Choose **Paste** from the **Edit** menu.

To copy an object or group of objects and paste in another screen:

1. Select the object or group of objects and choose **Copy** from the **Edit** menu. See "Selecting Multiple Objects" for instructions on grouping objects.
2. Open the screen to which to paste the object or objects.
3. Choose **Paste** from the **Edit** menu.

Delete

To delete an entry from a data field, highlight the data to delete and press the Backspace key.

To delete an object or group of objects from your screen:

1. Select the object group of objects to delete. See "Selecting Multiple Objects" for instructions on grouping objects.
2. Choose **Delete** from the **Edit** menu. The object or group of objects is removed from the screen and the Message bar displays "Delete performed."

SCREEN OBJECTS: DESCRIPTIONS AND PROPERTIES

Screen Builder features three types of objects:

- Analog – Objects that are linked to analog points (analog input, analog output, or pulse counters). Analog objects, like digital objects, mimic the activity of the analog points to which they are linked. An analog gauge object linked to a point that monitors a live gauge, which is reading a well level, will display the well's real-time level. When the well is at 45 feet, the analog object displays that reading.
- Digital – Objects that are linked to digital points (digital input or digital output). Digital objects mimic the activity, the state, of the digital points to which they are linked. If you have a digital point that tells you that a pump is running and that point is linked to a digital object, for example, a spinner, the spinner will rotate when the pump is on.
- Static – Objects that aren't linked to an active point. Static objects are used strictly for adding visual interest and depth to your screens. They can be used to create a background, a heading, or to add detail to your screen.

NOTE: The Pipe menu contains static and digital objects.

The following sections provide descriptions and examples of each object, and a list of its properties. The objects' properties can be used to alter its appearance and behavior (for example, to change an object's color or size, or link it to another screen or web page).

ANALOG SCREEN OBJECTS

BAR GRAPH (ANALOG)

Description	Example
Object resembling a bar graph that can be linked to an analog input (monitor) register. The bar fills as the register's value increases and empties as the value decreases. Variables, including bar color, background color, graph's low and high limits, and units to display (e.g., FT or PSI), can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	Analog value	Type a valid analog value in this field to see how the object will appear when the corresponding register reaches that value.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AI	Typical type of register for this object
OBJECT	BARGRAPH	Object's name
lowLimit	floating decimal value	The lowest limit (engineering value) – including resolution – that you want the system to display
highLimit	floating decimal value	The highest limit (engineering value) – including resolution – that you want the system to display
units	text string	Type of units to display, for example, FT, PSI, DegF
image	file name	File name for the image to overlay (place over) the bar graph. For example, an image of a chlorinator
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
vertical	V or H	Orientation of bar graph: V=vertical; H=horizontal
foregroundColor	color name or 6-character HEX	Color of the filled portion of the bar
backgroundColor	color name or 6-character HEX	Background color of the bar (unfilled portion)
showText	Y or N	Y displays the register's current value on the graph; N turns off this display
textColor	color name or 6-character HEX	Color of text for register's current value
showLimits	Y or N	Y displays the low and high limits on the graph; N turns off this display

Field Name	Value	Description
limitsColor	color name or 6-character HEX	Color of text for low and high limits
limitsPosition	L, R, T, B	Where limits should be displayed on graph: L=left, R=right, T=top, B=bottom

COLOR (ANALOG)

Description	Example
Rectangular object that can be linked to an analog input (monitor) register and whose color and text can change when the register's value changes. As the register's value increases or decreases, the object's color gradually shifts between the start color (the register's low limit) and the end color (the register's high limit). This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	Analog value	Type a valid analog value in this field to see how the object will appear when the corresponding register reaches that value.
ADDR	register	Register to which object is linked
XLOC	integer (pixel)	Horizontal position of object
YLOC	integer (pixel)	Vertical position of object
TYPE	AI	Typical type of register for this object
OBJECT	AACOLOR	Object's name
lowLimit	floating decimal value	The lowest limit (engineering value) – including resolution – that you want the system to display
highLimit	floating decimal value	The highest limit (engineering value) – including resolution – that you want the system to display
units	text string	Type of units to display, for example, FT, PSI, DegF
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
insetX	integer (pixels)	Left or right margin (depending on horizontal setting specified in justify) as measured from edge of object to label . If text label is right aligned, this is the size of the right margin. For a left aligned label, this is the size of the left margin.

Field Name	Value	Description
insetY	integer (pixels)	Top or bottom margin (depending on vertical setting specified in justify) as measured from edge of object to label . If label is top aligned, this is the size of the top margin. For a bottom aligned label, this is the size of the bottom margin.
justify	two-character text string	Vertical and horizontal alignment of the label with respect to the object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the text in the center [horizontally] and top [vertically] of the object.)
textColor	color name or 6-character HEX	Color of text
borderColor	color name or 6-character HEX	Color of object's border
drawBorder	Y or N	Y draws border around object; N turns off border
fontSize	integer	Font size to use for text
showText	Y or N	Y displays the register's current value on the graph; N turns off this display
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
drawOval	Y or N	Shape of object: Y=circle/oval; N=square/rectangle
startColor	color name or 6-character HEX	Color to assign to low limit value of object. As the register's value increases or decreases, the object's color gradually shifts between the startColor and the endColor, and vice versa.
endColor	color name or 6-character HEX	Color to assign to high limit value of object. As the register's value increases or decreases, the object's color gradually shifts between the startColor and the endColor, and vice versa.

CONTROL (ANALOG)

Description	Example
Object that can be used to control the value of an analog output (control) register. The object can be drawn horizontally or vertically. The high and low limits can appear at the top or bottom (for horizontal objects) or left or right (for vertical objects). The object fills left to right (or top to bottom) to show the register's current value. When the object is clicked, an editable field appears in the center of the object. Type the value you want to control the register to and press Enter.	

Field Name	Value	Description
Value	Analog value	Type a valid analog value in this field to see how the object will appear when the corresponding register reaches that value.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AO	Typical type of register for this object
OBJECT	ACONTROL	Object's name
lowLimit	floating decimal value	The lowest limit (engineering value) – including resolution – that you want the system to display
highLimit	floating decimal value	The highest limit (engineering value) – including resolution – that you want the system to display
units	text string	Type of units to display, for example, FT, PSI, DegF
image	file name	File name for the image to overlay (place over) the object. For example, an image of a chlorinator.
URL	N	Not applicable
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
vertical	V or H	Orientation of object: V=vertical; H=horizontal
foregroundColor	color name or 6-character HEX	Color of filled portion of the bar
backgroundColor	color name or 6-character HEX	Background color of bar (unfilled portion)
showText	Y or N	Y displays the register's current value on the bar; N turns off this display
textColor	color name or 6-character HEX	Color of text for register's current value
showLimits	Y or N	Y displays the low and high limits for the register on the bar; N turns off this display
limitsColor	color name or 6-character HEX	Color of text for low and high limits
limitsPosition	L, R, T, B	Location of limits display: For vertical bar, L=left, R=right; For horizontal bar, T=top, B=bottom
fontSize	integer	Font size to use for text
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text

DIAL (ANALOG)

Description	Example
Object that resembles the hand on a gauge. It can be linked to an analog input (monitor) register and paired with another object (e.g., RECTANGLE or OVAL) to create a custom gauge. The high and low degree limits of the dial as well as the high and low value limits to display can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	Analog value	Type a valid analog value in this field to see how the object will appear when the corresponding register reaches that value.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AI	Typical type of register for this object
OBJECT	DIAL	Object's name
lowLimit	floating decimal value	The lowest limit (engineering value) – including resolution – that you want the system to display
highLimit	floating decimal value	The highest limit (engineering value) – including resolution – that you want the system to display
lowDegrees	integer (0-360 degrees)	Starting point of needle's arc.
highDegrees	integer (0-360 degrees)	Ending point of needle's arc.
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
dialColor	color name or 6-character HEX	Color of needle
insideRadius	integer (pixels)	Radius of dial's inner arc; determines how close to the object's edge the needle begins.
lineWidth	integer (pixels)	Width of needle
clockwise	Y or N	Direction of needle's movement: Y=clockwise; N=counterclockwise.

GAUGE (ANALOG)

Description	Example
Object resembling a gauge that can be linked to an analog input (monitor) register and can display the register's current value in text and on the gauge's dial. The lowest and highest limits – including resolution – that should be displayed can be adjusted as well as the type of units (e.g., FT, PSI, DegF). This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	Analog value	Type a valid analog value in this field to see how the object will appear when the corresponding register reaches that value.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AI	Typical type of register for this object
OBJECT	GAUGE	Object's name
lowLimit	floating decimal value	The lowest limit (engineering value) – including resolution – that you want the system to display
highLimit	floating decimal value	The highest limit (engineering value) – including resolution – that you want the system to display
units	text string	Type of units to display, for example, FT, PSI, DegF
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
params	N	Not applicable

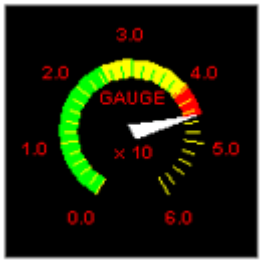
LED BAR (ANALOG)

Description	Example
Object similar to the BAR GRAPH except the graph is made up of individual bars that brighten or dim as the register's value increases and decreases. The bars can be color coded to indicate the register is in a safe range or a warning range (elevated but not yet in alarm), or has entered the alarm state. The high and low limits for each range and the colors associated with them can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	Analog value	Type a valid analog value in this field to see how the object will appear when the corresponding register reaches that value.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AI	Typical type of register for this object
OBJECT	LEDBAR	Object’s name
low	floating decimal value	Bar's low engineering value
warning	floating decimal value	Value at which you want to be alerted that the register is approaching the alarm state; the LEDs between the warning and the alarm limits will be displayed in the mediumColor
alarm	floating decimal value	Value at which the register enters an alarm state; the LEDs between the alarm and the high limits will be displayed in the highColor
high	floating decimal value	Bar's high engineering value
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
spacing	integer (pixels)	Amount of space between each LED
count	integer	Number of LEDs on the bar
backgroundColor	color name or 6-character HEX	Bar's background color
lowColor	color name or 6-character HEX	Color of bar's LEDs when register’s value is between the low and warning limits
mediumColor	color name or 6-character HEX	Color of bar's LEDs when register’s value is between the warning and alarm limits
highColor	color name or 6-character HEX	Color of bar's LEDs when register’s value is between the alarm and high limits
vertical	Y or N	Orientation of the bar: Y=vertical; N=horizontal
inverse	Y or N	Direction of the LED's values (low to high); Y displays the low value at the top of the bar (vertical bar) or on the right (horizontal bar); N displays the high value at the top of the bar (vertical bar) or on the right (horizontal bar)

Field Name	Value	Description
threshold	L, M, H	Bar's sensitivity to changes in the register's value; L (low) specifies that any time the register's value is in the range of the LED, the LED will illuminate; M (medium) specifies that the register's value will have to meet or exceed the midpoint of the LED's range before the LED will illuminate; H (high) specifies that the register's value will have to reach the maximum of the LED's range in order for the LED to illuminate.
barType	R, 3D, T, C	Shape and appearance of the LEDs; R=rectangular; 3D=three-dimensional, rectangular; T=triangular; C=circular

LED GAUGE (ANALOG)

Description	Example
Similar to the GAUGE except that the dial can be color coded to indicate that the value of the analog input register is in the safe range or warning range (elevated but not yet in alarm), or has entered the alarm state. The high and low limits for each range and the colors associated with them can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	Analog value	Type a valid analog value in this field to see how the object will appear when the corresponding register reaches that value.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AI	Typical type of register for this object
OBJECT	ANGAUGE	Object's name
low	integer	The gauge's low engineering value
warning	integer	The value at which you want to be alerted that the register is approaching the alarm state; the gauge's dial will display the mediumColor when the value is between the warning and alarm limits
alarm	integer	The value at which the register enters an alarm state; the gauge's dial will display the highColor when the value is between the alarm and high limits
high	integer	The gauge's high engineering value

Field Name	Value	Description
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
degrees	90, 180, 300, or 360 degrees	Measurement of the gauge's arc (90 for a quarter circle, 180 for a half circle, 300 for a nearly full circle, 360 for a full circle)
majorTicks	integer	Value of each of the gauge's major tick marks; also the value of the multiplier, which is displayed in the center of the gauge
decimal	integer	Number of decimal places to display for the gauge's labels
minorTicks	integer	Value of each of the gauge's minor tick marks (marks that appear between the major tick marks)
clockwise	Y or N	Direction of needle’s movement: Y=clockwise; N=counterclockwise
backgroundColor	color name or 6-character HEX	Color of gauge's background
lowColor	color name or 6-character HEX	Color of dial when register’s value is between the low and warning limits
mediumColor	color name or 6-character HEX	Color of dial when register’s value is between the warning and alarm limits
highColor	color name or 6-character HEX	Color of dial when register’s value is between the alarm and high limits
needleColor	color name or 6-character HEX	Color of needle
showLabels	Y or N	Y displays major tick value labels along outside of dial; N removes labels from display
textColor	color name or 6-character HEX	Color of the gauge's central text and major tick value labels
centralHeight	integer	Size of text displayed in gauge's center
labelHeight	integer	Size of text labels
centralText	text string	Text that will be displayed in the center of the gauge

LED TEXT (ANALOG)

Description	Example
LED object that can display the current value of an analog input (monitor) register. Properties such as the LED color, background color, decimal places to display, and displaying a + or - character to indicate positive and negative values can be adjusted. This object can also be used to create a slide-show affect by configuring it with a countdown time and a URL instead of an analog input register. The screen will automatically load the page specified in the URL field when the countdown timer has expired. This object can also be configured as a manual link to another screen or an HTML page.	

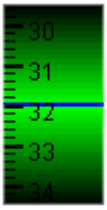
Field Name	Value	Description
Value	Analog value	Type a valid analog value in this field to see how the object will appear when the corresponding register reaches that value.
ADDR	register or STATIC	Register to which object is linked. Leave at default STATIC is this object is to be used as a countdown timer with a link to another screen or Web page.
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AI, ST	Typical type of register for this object
OBJECT	SEGLED	Object's name
low	integer	The lowest limit (engineering value) – including resolution – that you want the system to display
high	integer	The highest limit (engineering value) – including resolution – that you want the system to display
decimal	integer	Number of decimal places that the LED Text object is to display
signed	Y or N	Display a plus (+) or minus (-) to reflect positive and negative values. Y turns this option on.
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of text object
heightDim	integer (pixels)	Height (vertical size) of text object
backgroundColor	color name or 6-character HEX	Object's background color
digitColor	color name or 6-character HEX	Color of displayed text
countDown	integer (seconds)	Length of time before link (URL) is executed; enter STATIC in ADDR field to implement this feature

PANEL (ANALOG)

Description	Example
Object resembling a control panel display that can be linked to an analog input (monitor) register and can display the current value of the register. The low and high limits – including resolution – that should be displayed can be adjusted as well as the type of units (e.g., FT, PSI, DegF). This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	Analog value	Type a valid analog value in this field to see how the object will appear when the corresponding register reaches that value.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AI	Typical type of register for this object
OBJECT	PANEL	Object's name
lowLimit	floating decimal value	The lowest limit (engineering value) – including resolution – that you want the system to display
highLimit	floating decimal value	The highest limit (engineering value) – including resolution – that you want the system to display
units	text string	Type of units to display, for example, FT, PSI, DegF
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
params	N	Not applicable

ROTARY (ANALOG)

Description	Example
Object resembling a rotary gauge that can be linked to an analog input (monitor) register and can display the register's current value on the gauge's dial. Colors for the center of the rotary and the outside edges can be adjusted to give the object a curved look. The object can be drawn vertically or horizontally. Variables such as the range of values visible on the rotary's face, value of each of the gauge's major tick marks and minor tick marks can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	Analog value	Type a valid analog value in this field to see how the object will appear when the corresponding register reaches that value.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AI	Typical type of register for this object
OBJECT	ROTARY	Object's name
low	floating decimal value	Rotary's low engineering value (including resolution)
high	floating decimal value	Rotary's high engineering value (including resolution)
limitColor	color name or 6-character HEX	Color that appears at the outside edges of the rotary's gradient background
centerColor	color name or 6-character HEX	Color that appears in the center of the rotary's gradient background
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
visibleRange	integer	Range of values visible on the rotary's face at any one time
majorTicks	integer	Value of each of the gauge's major tick marks
minorTicks	integer	Value of each of the gauge's minor tick marks (tick marks that appear between the major tick marks)
horizontal	Y or N	Orientation of rotary: Y=horizontal; N=vertical
gradient	Y or N	Option for displaying a gradient background; Y for gradient background; N for transparent background
direction	Y or N	Direction that rotary's values are displayed (depends on orientation); Y for down or left (highest values are at the top or start on the left); N for up or right (highest values are at the bottom or start on the right)
leftTicks	Y or N	Placement of the rotary's tick marks (depends on rotary's orientation); Y for tick marks on the left or top of the rotary; N for tick marks on the right or bottom of the rotary
rotate	Y or N	Option for animating object (appears to turn, or spin, when the register's value changes). Y turns animation on; N turns off animation
needleColor	color name or 6-character HEX	Color of needle

SLIDER (ANALOG)

Description	Example
Similar to the CONTROL object except that the register is controlled by moving the slider. When the slider is moved, a value appears next to it to indicate the value that position represents. Variables such as orientation (horizontal or vertical), pointer and track style, colors (background, track, text), and text size can be adjusted.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AO	Typical type of register for this object
OBJECT	ASLIDER	Object's name
low	integer	Low engineering value (including resolution)
high	integer	High engineering value (including resolution)
decimal	integer	Number of decimal places to include in the text (value) display; this text appears next to the slider as the pointer is moved along the track
vertical	Y or H	Orientation: Y=vertical position; N=horizontal
URL	N	Not applicable
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
reverse	Y or N	Direction of pointer's movement; depends on object's orientation. Vertical slider: Y=bottom to top; N=top to bottom. Horizontal slider; Y=right to left; N=left to right.
showBackground	Y or N	Y to display a background behind the object; N for transparent background
backgroundColor	color name or 6-character HEX	Color of slider's background
backgroundStyle	R, L, F	Background style: R=raised, L=lowered; F=flat
pointerWidth	integer (pixels)	Width of pointer
pointerColor	color name or 6-character HEX	Color of pointer

Field Name	Value	Description
pointerStyle	L, R, U, D, B	Pointer's direction and style; depends on object's orientation. Vertical sliders: L for left or R for right; Horizontal sliders: U for up or D for down; B for box (can be used with either vertical or horizontal sliders)
trackStyle	S, M, L	Width of track (relative to overall width of object); S=small (narrow); M=medium; L=large (wide)
trackColor	color name or 6-character HEX	Color of track
textSize	integer	Size of the text used to display the value the register is being controlled to; text appears next to the slider as the pointer is moved along the track
textColor	color name or 6-character HEX	Color of the text used to display the value the register is being controlled to; text appears next to the slider as the pointer is moved along the track

TEXT (ANALOG)

Description	Example
Simple text object that can be linked to an analog input (monitor) register. The object's text can change to reflect the register's current value. Features, including the text to be displayed for each state, text color and size, and background color, can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	Analog value	Type a valid analog value in this field to see how the object will appear when the corresponding register reaches that value.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AI	Typical type of register for this object
OBJECT	AATEXT	Object's name
lowLimit	floating decimal value	The lowest limit (engineering value) – including resolution – that you want the system to display
highLimit	floating decimal value	The highest limit (engineering value) – including resolution – that you want the system to display
units	text string	Type of units to display, for example, FT, PSI, DegF
ALARM	N	Not applicable

Field Name	Value	Description
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of text object
heightDim	integer (pixels)	Height (vertical size) of text object
insetX	integer (pixels)	Left or right margin (depending on horizontal setting specified in justify) as measured from edge of text object to label . If label is right aligned, this is the size of the right margin. For a left aligned label, this is the size of the left margin.
insetY	integer (pixels)	Top or bottom margin (depending on vertical setting specified in justify) as measured from edge of text object to label . If label is top aligned, this is the size of the top margin. For a bottom aligned label, this is the size of the bottom margin.
justify	two-character text string	Vertical and horizontal alignment of the label with respect to the text object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the text in the center [horizontally] and top [vertically] of the object.)
textColor	color name or 6-character HEX	Color of text
backgroundColor	color name or 6-character HEX	Object's background color
drawBorder	Y or N	Y draws border around object; N turns off border
fontSize	integer	Font size to use for text
fillBackground	Y or N	Y fills text object with color specified in backgroundColor; N makes the object appear transparent
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
concatenate	N	Not applicable

TIME (ANALOG)

Description	Example
Object that can display the time of the workstation computer that is running the PMT software or can be linked to a logical register that calculates the number of seconds a logical input is in the ON state (for example, pump run time). When linked to this type of logical register, TIME will display (and continually and automatically update) the value of the specified logical register. The object can be configured to show date and time and can display one of four date formats. Characteristics such as background color and text color can also be adjusted.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	Logical register or STATIC	Logical register to which object is linked. This should be a logical register that calculates the length of time in seconds that a logical input, for example, PUMP RUN TIME, is in the ON state. Type STATIC in this field to have the object show the workstation's time of day.
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AI	Typical type of register for this object
OBJECT	AATIME	Object's name
showSeconds	Y or N	Y displays seconds; N turns off display.
showAMPM	Y or N	Y displays AM or PM; N turns off display.
dateFormat	N, S, M, L, F	Format of current workstation DATE; S for short (MM/DD/YY); M for medium (Mon DD, YY); L for long (Month DD, YYYY); F for full (Day of week, Month DD, YYYY); enter N to have object display current workstation TIME
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of time object
heightDim	integer (pixels)	Height (vertical size) of time object
insetX	integer (pixels)	Left or right margin (depending on horizontal setting specified in justify) as measured from edge of object to text string . If string is right aligned, this is the size of the right margin. For a left aligned string, this is the size of the left margin.
insetY	integer (pixels)	Top or bottom margin (depending on vertical setting specified in justify) as measured from edge of object to text string . If string is top aligned, this is the size of the top margin. For a bottom aligned string, this is the size of the bottom margin
justify	two-character text string	Vertical and horizontal alignment of the time string with respect to the time object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the time character string in the center [horizontally] and top [vertically] of the object.).

Field Name	Value	Description
textColor	color name or 6-character HEX	Color of time string (workstation time or date, or value of logical register).
backgroundColor	color name or 6-character HEX	Object's background color
drawBorder	Y or N	Y draws border around object; N turns off border
fontSize	integer	Font size to use for time character string.
fillBackground	Y or N	Y fills time object with color indicated in backgroundColor; N makes object transparent
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
concatenate	N	Not applicable

TIME CONTROL (ANALOG)

Description	Example
Similar to the TIME object except that TIME CONTROL can be linked to a logical analog input register that allows a user to enter the time when a certain action should occur (for example turning on an irrigation system). When the TIME CONTROL object is clicked, an editable box appears in the center of the object. The user enters the desired time (in hours and minutes) and presses the Enter key. When that time is reached, the control will occur. The TIME CONTROL object converts the time entered to seconds after midnight, which is easy to compare to the current time of day in the ladder logic that controls the irrigation system. For example, you want to control when an irrigation system comes on. In Logic Builder, logic is created where the time to control the event is provided by an analog input logical register (e.g., V_SPRINK_ON). The logical register is used in a comparison equation (when the current time is equal to the time specified by the logical register V_SPRINK_ON, the control register that regulates the irrigation system is activated). On the custom screen, the Analog TIME CONTROL object would be linked to the logical register V_SPRINK_ON.	

Field Name	Value	Description
Value	N/A	Not used for this object.
ADDR	Logical register or STATIC	Logical register to which object is linked. This should be a logical analog input register that allows a user to enter the time when a certain action should occur. Type STATIC in this field to have the object show the workstation's time of day.
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AO	Typical type of register for this object
OBJECT	CTRLTIME	Object's name

Field Name	Value	Description
showSeconds	Y or N	Y displays seconds; N turns off display
showAMPM	Y or N	Y displays AM or PM; N turns off display
INIT	N	Not applicable
ALARM	N	Not applicable
URL	N	Not applicable
widthDim	integer (pixels)	Width (horizontal size) of time control object
heightDim	integer (pixels)	Height (vertical size) of time control object
insetX	integer (pixels)	Left or right margin (depending on horizontal setting specified in justify) as measured from edge of object to text string . If string is right aligned, this is the size of the right margin. For a left aligned string, this is the size of the left margin.
insetY	integer (pixels)	Top or bottom margin (depending on vertical setting specified in justify) as measured from edge of object to text string . If string is top aligned, this is the size of the top margin. For a bottom aligned string, this is the size of the bottom margin.
justify	two-character text string	Vertical and horizontal alignment of the time string with respect to the time control object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the time character string in the center [horizontally] and top [vertically] of the object.).
textColor	color name or 6-character HEX	Color of time character string
backgroundColor	color name or 6-character HEX	Object's background color
drawBorder	Y or N	Y draws border around object; N turns off border
fontSize	integer	Font size to use for time character string
fillBackground	Y or N	Y fills time control object with color indicated in backgroundColor; N makes object transparent
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
concatenate	N	Not applicable

TREND (ANALOG)

Description	Example
Object that can be linked to an analog input (monitor) register and that can display changes in the register's value via a trend line. Variables, including the range of the trend (high and low limits), the span of time (in minutes) to display, and the color of the line, can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	Analog value	Type a valid analog value in this field to see how the object will appear when the corresponding register reaches that value.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	AI	Typical type of register for this object
OBJECT	ATREND	Object's name
lowLimit	floating decimal value	The lowest limit (engineering value) – including resolution – that you want the system to display
highLimit	floating decimal value	The highest limit (engineering value) – including resolution – that you want the system to display
minutes	integer (minutes)	Data width; span of time over which you want to see data reported
color	color name or 6-character HEX	Color of trend line
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
smooth	Y or N	Y smooths the trend; changes are displayed in a more gradual manner. N turns off the smooth option.

DIGITAL SCREEN OBJECTS

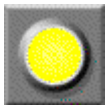
3D RECTANGLE (DIGITAL)

Description	Example
Rectangular object with a 3-dimensional border that can be linked to a digital input (monitor) register. The rectangle's color and text can change when the register's status changes. The ON and OFF colors and text can be specified as well as the text color and size, and border color and size. The color and width of the border can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0, 1	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	AD3DRECT	Object's name
ON (1)	label or N	Text to display during ON state; system automatically fills this field based on label configured for register. Enter N (uppercase) to turn off text display.
OFF (0)	label or N	Text to display during OFF state; system automatically fills this field based on label configured for register. Enter N (uppercase) to turn off text display.
onColor	color name or 6-character HEX	Color of object during ON state
offColor	color name or 6-character HEX	Color of object during OFF state
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
insetX	integer (pixels)	Left or right margin (depending on horizontal setting specified in justify) as measured from edge of object to label
insetY	integer (pixels)	Top or bottom margin (depending on vertical setting specified in justify) as measured from edge of object to label
justify	2-character string	Vertical and horizontal alignment of the label with respect to the object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the text in the center [horizontally] and top [vertically] of the object.)
borderColor	color name or 6-character HEX	Color of object's border
textColor	color name or 6-character HEX	Color of text
drawBorder	Y or N	Y draws border around object; N turns off border
fontSize	integer	Font size to use for text

Field Name	Value	Description
fillBackground	Y or N	Y fills object with color specified in backgroundColor; N makes object transparent
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
concatenate	N	Not applicable
showText	Y or N	Y turns on text display; N turns off text display
borderWidth	integer (pixels)	Width of border (in pixels)
drawRaised	Y or N	Y makes object appear raised; N makes object look lowered (pressed in)

4-STATE GRAPHIC (DIGITAL)

Description	Example
Object that can be linked to a four-state digital register. Similar to a 4-STATE RECTANGLE, except that the object can be configured to display a different .gif image for each of its four states as well as the initial state. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0, 1, 2, 3	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	AD4IMAGE	Object's name
image0	file name	Image to display during 0 (zero) state – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). N turns off image display.
image1	file name	Image to display during 1 (one) state – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). N turns off image display.
INIT	file name	Image to display during INIT state (register isn't offline, but doesn't yet have status) – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). N turns off image display.
ALARM	N	For future use.

Field Name	Value	Description
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
image2	file name	Image to display during two (2) state – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). N turns off image display.
image3	file name	Image to display during 3 (three) state – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). N turns off image display.

4-STATE RECTANGLE (DIGITAL)

Description	Example
Rectangular object that can be linked to a four-state digital register. This object can display one of four states: 0 (zero), 1 (one), 2 (two) or 3 (three). The object can be configured to display a different color for each of the four states. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0, 1, 2, 3	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	AD4RECT	Object's name
color0	color name or 6-character HEX	Color of object during 0 (zero) state
color1	color name or 6-character HEX	Color of object during 1 (one) state
color2	color name or 6-character HEX	Color of object during 2 (two) state
color3	color name or 6-character HEX	Color of object during 3 (three) state
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
borderWidth	integer (pixels)	Width of border

Field Name	Value	Description
drawRaised	Y or N	Y makes object appear raised; N makes object look lowered (pressed in)

4-STATE TEXT (DIGITAL)

Description	Example
Similar to a 4-STATE RECTANGLE, except that the object can be configured to display a different text label for each of its four states. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0, 1, 2, 3	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	AD4TEXT	Object's name
label0	label	Text to display during 0 (zero) state
label1	label	Text to display during 1 (one) state
label2	label	Text to display during 2 (two) state
label3	label	Text to display during 3 (three) state
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
insetX	integer (pixels)	Left or right margin (depending on horizontal setting specified in justify) as measured from edge of object to label
insetY	integer (pixels)	Top or bottom margin (depending on vertical setting specified in justify) as measured from edge of object to label
justify	two-character text string	Vertical and horizontal alignment of the label with respect to the object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the text in the center [horizontally] and top [vertically] of the object.)

Field Name	Value	Description
fontSize	integer	Font size to use for text
textColor	color name or 6-character HEX	Color of text
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text

ANIMATION (DIGITAL)

Description	Example
Animated object that can be linked to a digital input (monitor) register. The object is animated when the register is in the ON state. The animation is accomplished using a series of .gif images each of which is drawn slightly different to mimic movement when played in sequence. Different .gif files can be specified for the OFF and initial states. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0, 1	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	ANIMATE	Object's name
ON (1)*	file name	Base image to display during ON state – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). Enter N to turn off image display.
OFF (0)*	color name or 6-character HEX	Base image to display during OFF state – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). Enter N to turn off image display.
INIT**	color name or 6-character HEX	Base image to display during initial state (register isn't offline, but doesn't yet have status – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). Enter N to turn off image display.
ALARM*	N	For future use
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
onFrames*	integer	Number of frames to use for ON state

Field Name	Value	Description
offFrames*	integer	Number of frames to use for OFF state
alarmFrames*	N	For future use

* The ON and OFF states require a base image name and a frame count. Each frame count is appended to the base image name to form the name of the image to draw in each frame. For example, if the ON state has a base image named "pump" and a frame count of 4, then the four images pump1.gif, pump2.gif, pump3.gif, and pump4.gif are used to create the animation for the ON state. All of these image file must be available in images subdirectory (see Choosing Images on page 117 for more information).

If the number of frames entered in these fields exceeds the number of files that exist, the system uses the files that are available and inserts a time delay for each file not available. In the above example, if a frame count of 6 was entered, the four "pump" images would be displayed and then there would be a length of time (a second or two) when the image was not visible on the screen.

**The INIT state is not animated.

ARROW (DIGITAL)

Description	Example
Arrow-shaped object that can be linked to a digital input (monitor) register and whose color can change to reflect the register's current status. The ON and OFF colors, arrow direction, and tail size can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0, 1	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	DARROW	Object's name
direction	N, E, W, S	Direction arrow is pointing; N (up), E (right), W (left), S (down)
OFF (0)	N	Not applicable
drawBorder	Y or N	Y draws border around object; N turns off border
tailSize	S, M, L, N	Size of arrow's tail; S (small), M (medium), L (large), N (no tail)

Field Name	Value	Description
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
onColor	color name or 6-character HEX	Color of arrow during ON state
offColor	color name or 6-character HEX	Color of arrow during OFF state
borderColor	color name or 6-character HEX	Color of arrow's border

CONTROL RECTANGLE (DIGITAL)

Description	Example
Rectangular object that can be linked to a digital output (control) register. This object can be used to force a register to a specific state by clicking the object (turning the button on or off). The object's color and text can change when the register's state changes. The ON and OFF colors and text can be specified as well as the text color and size, and border color and size.	

Field Name	Value	Description
Value	0, 1	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DO	Typical type of register for this object
OBJECT	DCTRLREC	Object's name
ON (1)	label or N	Text to display during ON state; system automatically fills this field based on label configured for register in I/O Builder. Enter N (uppercase) to turn off text display.
OFF (0)	label or N	Text to display during OFF state; system automatically fills this field based on label configured for register in I/O Builder. Enter N (uppercase) to turn off text display.
onColor	color name or 6-character HEX	Color of object during ON state
offColor	color name or 6-character HEX	Color of object during OFF state
URL	N	Not applicable
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object

Field Name	Value	Description
insetX	integer (pixels)	Left or right margin (depending on horizontal setting specified in justify) as measured from edge of object to label
insetY	integer (pixels)	Top or bottom margin (depending on vertical setting specified in justify) as measured from edge of object to label
justify	2-character string	Vertical and horizontal alignment of the label with respect to the object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the text in the center [horizontally] and top [vertically] of the object.)
borderColor	color name or 6-character HEX	Color of object's border
textColor	color name or 6-character HEX	Color of text
drawBorder	Y or N	Y draws border around object; N turns off border
fontSize	integer	Font size to use for text
fillBackground	Y or N	Y fills object with color specified in onColor/offColor; N makes the object transparent
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
concatenate	N	Not applicable
showText	Y or N	Y turns on display of ON/OFF labels; N turns off display.
borderWidth	integer (pixels)	Width of border
drawRaised	Y or N	Y makes object appear raised; N makes object look lowered (pressed in)

GRAPHIC (DIGITAL)

Description	Example
Object that can be linked to a digital input (monitor) register and can display up to three distinct .gif images to reflect the register's current state (ON, OFF, and initialized). The object can also be configured to display an ON and OFF text label. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0, 1	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked

Field Name	Value	Description
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	DIGITAL	Object's name
ON (1)	file name or N	Image to display during ON state – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). N turns off image display.
OFF (0)	file name or N	Image to display during OFF state – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). N turns off image display.
INIT	file name or N	Image to display during initial state (register isn't offline, but doesn't yet have status) – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). N turns off image display.
ALARM	N	For future use
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
onLabel	text or N	Text to display during ON state; N turns off text display
offLabel	text or N	Text to display during OFF state; N turns off text display
showText	Y or N	Y turns on display of onLabel/offLabel; N turns off display.
textColor	color name or 6-character HEX	Color of text
fontSize	integer	Font size to use for text

GRAPHIC CONTROL (DIGITAL)

Description	Example
Object that can be linked to a digital output (control) register and can display a different image for each of its four states (on, off, and initial). This object can be used to force a register to a specific state by clicking the object. Options such as the images to display for each state, and the color and position of the label can be adjusted.	

Field Name	Value	Description
Value	0, 1	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DO	Typical type of register for this object
OBJECT	DCONTROL	Object's name
ON (1)	file name or N	Image to display during ON state – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). Enter N to turn off image display.
OFF (0)	file name or N	Image to display during OFF state – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). Enter N to turn off image display.
INIT	file name or N	Image to display during initial state (register isn't offline, but doesn't yet have status) – .gif file is assumed; file must be available in images subdirectory (see "Choosing Images" on page 117 for more information). Enter N to turn off image display.
ALARM	N	For future use
URL	N	Not applicable
onLabel	text or N	Text to display during ON state; enter N to turn off text display
offLabel	text or N	Text to display during OFF state; enter N to turn off text display
showText	Y or N	Y turns on display of ON/OFF label; N turns off display.
textColor	color name or 6-character HEX	Color of text
fontSize	integer	Font size to use for text
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
insetX	integer (pixels)	Left or right margin (depending on horizontal setting specified in justify) as measured from edge of object to label
insetY	integer (pixels)	Top or bottom margin (depending on vertical setting specified in justify) as measured from edge of object to label

Field Name	Value	Description
justify	2-character string	Vertical and horizontal alignment of the label with respect to the object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the text in the center [horizontally] and top [vertically] of the object.)

OVAL (DIGITAL)

Description	Example
Round object that can be linked to a digital input (monitor) register and whose color and text can change to reflect the register's current status (ON or OFF). By manipulating the height and the width, the object can be made to have a more circular or more oval shape. The object can be drawn with or without a border. Features such as border color and size, and text color and size can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0, 1	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	ADOVAL	Object's name
ON (1)	label or N	Text to display during ON state; system automatically fills this field based on label configured for register in I/O Builder. Enter N (uppercase) to turn off text display.
OFF (0)	label or N	Text to display during OFF state; system automatically fills this field based on label configured for register in I/O Builder. Enter N (uppercase) to turn off text display.
onColor	color name or 6-character HEX	Color of object during ON state
offColor	color name or 6-character HEX	Color of object during ON state
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object

Field Name	Value	Description
heightDim	integer (pixels)	Height (vertical size) of object
insetX	integer (pixels)	Left or right margin (depending on horizontal setting specified in justify) as measured from edge of object to label
insetY	integer (pixels)	Top or bottom margin (depending on vertical setting specified in justify) as measured from edge of object to label
justify	2-character string	Vertical and horizontal alignment of the label with respect to the object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the text in the center [horizontally] and top [vertically] of the object.)
borderColor	color name or 6-character HEX	Color of object's border
textColor	color name or 6-character HEX	Color of text
drawBorder	Y or N	Y draws border around object; N turns off border
fontSize	integer	Font size to use for text
fillBackground	Y or N	Y fills object with color specified in backgroundColor; N makes object transparent
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
concatenate	N	Not applicable
showText	Y or N	Y turns on text display; N turns off text display
borderWidth	integer (pixels)	Width of border

RECTANGLE (DIGITAL)

Description	Example
Rectangular object that can be linked to a digital input (monitor) register and whose color and text can change when the register's status changes. The ON and OFF colors and text can be specified as well as the text color and size, and border color and size. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0, 1	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object

Field Name	Value	Description
TYPE	DI	Typical type of register for this object
OBJECT	ADRECT	Object's name
ON (1)	label or N	Text to display during ON state; system automatically fills this field based on label configured for register in I/O Builder. Enter N (uppercase) to turn off text display.
OFF (0)	label or N	Text to display during OFF state; system automatically fills this field based on label configured for register in I/O Builder. Enter N (uppercase) to turn off text display.
onColor	color name or 6-character HEX	Color of object during ON state
offColor	color name or 6-character HEX	Color of object during OFF state
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
insetX	integer (pixels)	Left or right margin (depending on horizontal setting specified in justify) as measured from edge of object to label
insetY	integer (pixels)	Top or bottom margin (depending on vertical setting specified in justify) as measured from edge of object to label
justify	2-character string	Vertical and horizontal alignment of the label with respect to the object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the text in the center [horizontally] and top [vertically] of the object.)
borderColor	color name or 6-character HEX	Color of object's border
textColor	color name or 6-character HEX	Color of text
drawBorder	Y or N	Y draws border around object; N turns off border
fontSize	integer	Font size to use for text
fillBackground	Y or N	Y fills rectangle with color specified in onColor/offColor; N makes the object appear transparent
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
concatenate	N	Not applicable
showText	Y or N	Y turns on text display; N turns off text display

Field Name	Value	Description
borderWidth	integer (pixels)	Width of border (in pixels)

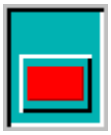
ROUND RECTANGLE (DIGITAL)

Description	Example
Similar to the RECTANGLE except that it features rounded corners. The amount of curve at the corners can be adjusted to make the object appear more square or more round. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0, 1	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	ADRRECT	Object's name
ON (1)	label or N	Text to display during ON state; system automatically fills this field based on label configured for register in I/O Builder. Enter N (uppercase) to turn off text display.
OFF (0)	label or N	Text to display during OFF state; system automatically fills this field based on label configured for register in I/O Builder. Enter N (uppercase) to turn off text display.
onColor	color name or 6-character HEX	Color of object during ON state
offColor	color name or 6-character HEX	Color of object during OFF state
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
insetX	integer (pixels)	Left or right margin (depending on horizontal setting specified in justify) as measured from edge of object to label
insetY	integer (pixels)	Top or bottom margin (depending on vertical setting specified in justify) as measured from edge of object to label

Field Name	Value	Description
justify	2-character string	Vertical and horizontal alignment of the label with respect to the object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the text in the center [horizontally] and top [vertically] of the object.)
borderColor	color name or 6-character HEX	Color of object's border
textColor	color name or 6-character HEX	Color of text
drawBorder	Y or N	Y draws border around object; N turns off border
fontSize	integer	Font size to use for text
fillBackground	Y or N	Y fills object with color specified in onColor/offColor; N makes the object transparent
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
concatenate	N	Not applicable
showText	Y or N	Y turns on text display; N turns off text display
borderWidth	integer (pixels)	Width of border
radius	integer	Larger numbers make the rectangle's corners appear rounder; the image is more circular. Smaller numbers make the rectangle's corners appear sharper; the image is more angular.

SWITCH (DIGITAL)

Description	Example
Object resembling a switch (rocker-, lever-, or slider-type switch) that can be linked to a digital output (control) register. This object can be used to force a register to a specific state by clicking the object (turning the switch on or off). Options such as switch type and color, LED display, and switch border can be adjusted.	 (Lever)  (Rocker)  (Slider)

Field Name	Value	Description
Value	0, 1	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DO	Typical type of register for this object
OBJECT	DSWITCH	Object's name
switchType	ROCKER, LEVER, SLIDER	Type of switch to display
switchColor	color name or 6-character HEX	Color of switch
drawBackground	Y or N	Y to draw switch's background; N for transparent background
borderColor	color name or 6-character HEX	Color of switch's border (ROCKER switch only)
URL	N	Not applicable
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
drawLED	Y or N	Y to include an LED that changes color depending on the register's state; N to turn off LED display
onColor	color name or 6-character HEX	Color of LED during ON state
offColor	color name or 6-character HEX	Color of LED during OFF state

TEXT (DIGITAL)

Description	Example
Simple text object that can be linked to a digital input (monitor) register. The object's text can change to reflect the register's current state (ON or OFF). Features, including the text to be displayed for each state, text color and size, and background color, can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0, 1	Type a valid value in this field to see how the object will appear when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	ADTEXT	Object's name
ON (1)	label or N	Text to display during ON state; system automatically fills this field based on label configured for register in I/O Builder. Enter N (uppercase) to turn off text display.
OFF (0)	label or N	Text to display during OFF state; system automatically fills this field based on label configured for register in I/O Builder. Enter N (uppercase) to turn off text display.
INIT	N	Not applicable
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
insetX	integer (pixels)	Left or right margin (depending on horizontal setting specified in justify) as measured from edge of object to label
insetY	integer (pixels)	Top or bottom margin (depending on vertical setting specified in justify) as measured from edge of object to label

Field Name	Value	Description
justify	two-character text string	Vertical and horizontal alignment of the label with respect to the object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the text in the center [horizontally] and top [vertically] of the object.)
textColor	color name or 6-character HEX	Color of text
backgroundColor	color name or 6-character HEX	Object's background color
drawBorder	Y or N	Y draws border around object; N turns off border
fontSize	integer	Font size to use for text
fillBackground	Y or N	Y fills object with color specified in backgroundColor; N makes the object transparent
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
concatenate	N	Not applicable

PIPE SCREEN OBJECTS

DIGITAL ELBOW (PIPE)

Description	Example
Object resembling a pipe elbow that can be linked to a digital input (monitor) register. The object can be configured with a different color for each of its possible states -- on, off, and initial (register isn't off line, but doesn't yet have status). The orientation of the elbow (north-east, north-west, south-east, or south-west) can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0 or 1	Enter a value in this field to see how the object will look when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	ADELBOW	Object's name
ON (1)	color name or 6-character HEX	Color that appears in the center of the elbow during ON state

Field Name	Value	Description
OFF (0)	color name or 6-character HEX	Color that appears in the center of the elbow during OFF state
INIT	color name or 6-character HEX	Color that appears in the center of the elbow during the initial state
backgroundColor	color name or 6-character HEX	Color of pipe
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
diameter	integer (pixels)	Diameter of pipe elbow
quadrant	SE, SW, NE, NW	Location of the elbow's outside curve (SE=bottom right, SW=bottom left, NE=top right, NW=top left)

DIGITAL PIPE (PIPE)

Description	Example
Object resembling a length of pipe that can be linked to a digital input (monitor) register. The object can be configured with a different color for each of its possible states -- on, off, and initial (register isn't off line, but doesn't yet have status). The pipe can be drawn with joints at one or both ends, or can be drawn with no joints. The orientation of the pipe (horizontal or vertical) can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0 or 1	Enter a value in this field to see how the object will look when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	ADPIPE	Object’s name
ON (1)	color name or color name or 6-character HEX	Color that appears in the center of the pipe during ON state
OFF (0)	color name or color name or 6-character HEX	Color that appears in the center of the pipe during OFF state
INIT	color name or color name or 6-character HEX	Color that appears in the center of the pipe during the initial state

Field Name	Value	Description
backgroundColor	color name or color name or 6-character HEX	Color of pipe
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
vertical	Y or N	Orientation of pipe: Y=vertical; N=horizontal
northEast	Y or N	Add junction: Type Y to draw a junction at the north (top) end of vertical pipe, or east (left) end of horizontal pipe. Type N for no junction.
southWest	Y or N	Add junction: Type Y to draw a junction at the south (bottom) end of vertical pipe, or at the west (right) end of horizontal pipe. Type N for no junction.

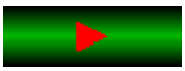
GRADIENT ELBOW (PIPE)

Description	Example
Static object (cannot be linked to a register) that can be paired with the GRADIENT PIPE. Features such as the color, size, and orientation of the pipe can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	GELBOW	Object's name
ON (1)	N	Not applicable
OFF (0)	N	Not applicable
INIT	N	Not applicable
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object

Field Name	Value	Description
heightDim	integer (pixels)	Height (vertical size) of object
foregroundColor	color name or 6-character HEX	Color of center, or bright, part of pipe
backgroundColor	color name or 6-character HEX	Color of outside, or dark, part of pipe
diameter	integer (pixels)	Diameter of pipe elbow
quadrant	SE, SW, NE, NW	Location of the elbow's outside curve (SE=bottom right, SW=bottom left, NE=top right, NW=top left)

GRADIENT PIPE (PIPE)

Description	Example
Animated pipe object that can be linked to a digital input (monitor) register. The object is darker at the edges to make it appear three-dimensional. When the register is on, the arrows in the pipe are animated to mimic movement. The object can be configured to show static arrows when the register is off. Features such as the color of the pipe and the arrows, and the size and orientation of the pipe can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0 or 1	Enter a value in this field to see how the object will look when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	ANIMPIPE	Object’s name
ON (1)	N	Not applicable
foregroundColor	color name or 6-character HEX	Color of center (or bright) part of pipe
backgroundColor	color name or 6-character HEX	Color of outside (or dark) part of pipe
arrowColor	color name or 6-character HEX	Color of pipe's animated arrows
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object

Field Name	Value	Description
showAnimation	Y or N	Y displays animated arrows (if object is linked to a register) or static arrows (if object is not linked to a register); N displays static arrows (if object is linked to a register) or no arrows (if object is not linked to a register)
vertical	Y or N	Orientation of pipe; Y=vertical; N=horizontal
direction	Y or N	Direction that arrows point and move when animated; depends on orientation: Y for arrows that move left (horizontal) or down (vertical); N for arrows that move right (horizontal) or up (vertical)
arrowType	S, L, or T	Size and characteristics of arrows; S for short arrows without tails; L for long arrows without tails; T for short arrows with tails
offArrow	Y or N	"Off" arrows are arrows that are a darker shade of the color that is specified in arrowColor. • If the object is linked to a register and animateArrows is enabled, Y displays "off" arrows between the animated arrows when the register is ON and static "off" arrows when the register is OFF • If the object is linked to a register and animateArrows is disabled, static "off" arrows are displayed when the register is off. • If the object is not linked to a register (STATIC), Y displays "off" arrows between the animated arrows • N turns off the display of "off" arrows when the object is linked to register *or* is STATIC

SPINNER (PIPE)

Description	Example
Animated object that can be linked to a digital input (monitor) register. When the register is on, the spinner rotates. The spinner can be configured with a different color for each of its possible states -- on, off, and initial (register isn't off line, but doesn't yet have status). The spin speed can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0, 1	Enter a value in this field to see how the object will look when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked

Field Name	Value	Description
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	SPINNER	Object's name
ON (1)	color name or 6-character HEX	Color of spinner during ON state
OFF (0)	color name or 6-character HEX	Color of spinner during OFF state
INIT	color name or 6-character HEX	Color of spinner during initial state (register isn't offline, but doesn't yet have status)
ALARM	N	For future use
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of spinner
heightDim	integer (pixels)	Height (vertical size) of spinner
spinColor	color name or 6-character HEX	Base color of spinner
degree	integer (degrees)	Angle at which spinner rotates (must divide evenly into 180). Low numbers make the spinner appear to be rotating slowly; high numbers make the spinner appear to be rotating faster.
flashAlarm	N	For future use
invertState	Y or N	Invert logic: Y inverts logic of 0 and 1 states – spinner does *not* rotate when register is ON

STATIC ELBOW (PIPE)

Description	Example
Static object (cannot be linked to a register) resembling a length of pipe. The orientation of the elbow (north-east, north-west, south-east, or south-west) and the color of the elbow can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object

Field Name	Value	Description
OBJECT	ELBOW	Object's name
ON (1)	N	Not applicable
OFF (0)	N	Not applicable
INIT	N	Not applicable
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
foregroundColor	color name or 6-character HEX	Color that appears in the center of the pipe
backgroundColor	color name or 6-character HEX	Color of pipe elbow
diameter	integer (pixels)	Diameter of pipe elbow
quadrant	SE, SW, NE, NW	Location of the elbow's outside curve (SE=bottom right, SW=bottom left, NE=top right, NW=top left)

STATIC PIPE (PIPE)

Description	Example
Static object (cannot be linked to a register) resembling a length of pipe. The pipe can be drawn with joints at one or both ends, or can be drawn with no joints. The orientation of the pipe (horizontal or vertical) and the color of the pipe can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	SPIPE	Object's name
ON (1)	N	Not applicable
OFF (0)	N	Not applicable
INIT	N	Not applicable
ALARM	N	Not applicable

Field Name	Value	Description
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
foregroundColor	color name or 6-character HEX	Color that appears in the center of the pipe
backgroundColor	color name or 6-character HEX	Color of the shaded portion of the pipe
vertical	Y or N	Y to draw pipe vertically; N to leave horizontally
northEast	Y or N	Add junction: Type Y to draw a junction at the north (top) end of vertical pipe, or east (left) end of horizontal pipe. Type N for no junction.
southWest	Y or N	Add junction: Type Y to draw a junction at the south (bottom) end of vertical pipe, or at the west (right) end of horizontal pipe. Type N for no junction.

VALVE (PIPE)

Description	Example
Object resembling a pipe valve that can be linked to a digital input (monitor) register. The valve can be configured to change color when the register's state changes. Features such as the ON and OFF color, and the size and orientation of the valve can be adjusted. The valve can be drawn with or without a handle. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	0 or 1	Enter a value in this field to see how the object will look when its corresponding register is in the specified state.
ADDR	register	Register to which object is linked
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	DI	Typical type of register for this object
OBJECT	DVALVE	Object's name
vertical	Y or N	Orientation of pipe: Y=vertical; N=horizontal
drawHandle	Y or N	Add handle: Y=handle on the valve; N=no handle
drawBorder	Y or N	Y draws border around object; N turns off border
borderColor	color name or 6-character HEX	Color of valve's border (drawBorder must be set to Y)

Field Name	Value	Description
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of valve
heightDim	integer (pixels)	Height (vertical size) of valve
onColor	color name or 6-character HEX	Color of valve when register is in the ON state
offColor	color name or 6-character HEX	Color of valve when register is in the OFF state

STATIC SCREEN OBJECTS

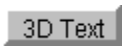
3D RECTANGLE (STATIC)

Description	Example
Rectangular object that features a three-dimensional border. The color and width of the border can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	S3DRECT	Object's name
ON (1)	N	Not applicable
OFF (0)	N	Not applicable
INIT	N	Not applicable
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
borderWidth	integer (pixels)	Width of border
borderColor	color name or 6-character HEX	Color of object's border
backgroundColor	color name or 6-character HEX	Object's background color

Field Name	Value	Description
fillBackground	Y or N	Y fills object with color specified in backgroundColor; N makes object transparent
drawBorder	Y or N	Y draws border around object; N turns off border
drawRaised	Y or N	Y makes object appear raised; N makes object look lowered (pressed in)

3D TEXT (STATIC)

Description	Example
Simple text object that features a three-dimensional border. This object can be used as a label or a short block of text. Up to 4 lines of text with up to 12 characters on each line can be accommodated. The color and width of the border can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	S3DTEXT	Object's name
line 1	text string	First label to display; up to 12 characters
line 2	text string	Second label to display; up to 12 characters
line 3	text string	Third label to display; up to 12 characters
line 4	text string	Fourth label to display; up to 12 characters
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
insetX	integer (pixels)	Left or right margin (depending on the horizontal setting specified in justify) as measured from edge of object to label
insetY	integer (pixels)	Top or bottom margin (depending on the vertical setting specified in justify) as measured from edge of object to label
justify	2-character text string	Vertical and horizontal alignment of the label with respect to the object; horizontal alignment is given

Field Name	Value	Description
		first. Horizontal values: C=center; R=right; L=left. Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the text in the center [horizontally] and top [vertically] of the object.)
textColor	color name or 6-character HEX	Color of text
backgroundColor	color name or 6-character HEX	Object's background color
drawBorder	Y or N	Y draws border around object; N turns off border
fontSize	integer	Font size to use for text
fillBackground	Y or N	Y fills object with color specified in backgroundColor; N makes object transparent
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
concatenate	Y or N	Y links the text strings in lines 1-4 to form one line of text; N places text in lines 1-4 on separate lines
borderColor	color name or 6-character HEX	Color of border
borderWidth	integer (pixels)	Width of border
drawRaised	Y or N	Y makes object appear raised; N makes object look lowered (pressed in)

BANNER TEXT (STATIC)

Description	Example
Text object resembling a scrolling LED marquee that accepts up to 90 text characters. Properties such as the LED color, background color, and scrolling speed can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	MATRLED	Object's name
LEDcolor	color name or 6-character HEX	Color of LEDs
backgroundColor	color name or 6-character HEX	Color of banner's background
speed	integer	Banner's scrolling speed
marquee	Y or N	Y causes banner to behave like scrolling marquee; N turns off marquee effect

Field Name	Value	Description
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
textLine	text string	Text that will appear on banner (up to 90 characters)

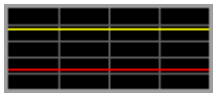
GRADIENT (STATIC)

Description	Example
Object that features a background that gradates (or shifts) from one color to another. This object can be used to give shading and depth to items such as tanks, pipes, or the sky. GRADIENT can be rectangular or oval in shape. The beginning and ending colors and the direction of the gradient (top to bottom; left to right) can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	GRADIENT	Object's name
fillRectangle	Y or N	Option to fill gradient from center out to edges; Y enables this option; N causes object to use either the direction specified in the direction field, or to use the fillOval option
fillOval	Y or N	Option to make object circular and fill from center out to edges; Y enables this option; N causes object to use either the direction specified in the direction field, or to use the fillRectangle option
INIT	N	Not applicable
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object

Field Name	Value	Description
direction	N, S, E, W, NE, NW, SE, SW	Direction of color shift; startColor begins at the opposite coordinate and gradually changes to the endColor in the direction specified here; this setting has no affect if fillRectangle or fillOval have been enabled
endColor	color name or 6-character HEX	Ending color of color shift
startColor	color name or 6-character HEX	Starting, or initial, color of color shift

GRID (STATIC)

Description	Example
Rectangular object that has a grid as its background and also features high and low limit lines. This object is especially useful as a background for an Analog TREND. Features including the value of the low and high limit lines, the object's orientation (vertical or horizontal), and the color of the limit lines can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	SGRID	Object’s name
horizontal	integer	Number of horizontal grid segments (not grid lines)
vertical	integer	Number of vertical grid segments (not grid lines)
limitLine1	integer (0-100 percent)	Distance between limitLine1 and grid’s lower edge
limitLine2	integer (0-100 percent)	Distance between limitLine2 and grid's lower edge
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
borderWidth	integer (pixels)	Width of border
borderColor	color name or 6-character HEX	Color of border
backgroundColor	color name or 6-character HEX	Object’s background color
fillBackground	Y or N	Y fills object with color specified in backgroundColor;

Field Name	Value	Description
		N makes the object's background transparent
drawBorder	Y or N	Y draws border around object; N turns off border
gridColor	color name or 6-character HEX	Color of grid lines
oneColor	color name or 6-character HEX	Color of limitLine1
twoColor	color name or 6-character HEX	Color of limitLine2

IMAGE (STATIC)

Description	Example
This object can display any .gif file stored in the images directory (usually C:\Program Files\PMT2\Images). It can be configured as a link to another screen or an HTML page.	

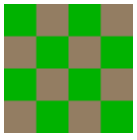
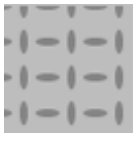
Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	IMAGE	Object's name
ON (1)	N	Not applicable
image	file name or N	File name of image to display (.gif file is assumed). file must be available in images subdirectory (see "Choosing Images" on page 117 for more information).
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)

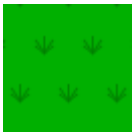
OVAL (STATIC)

Description	Example
A building block object similar to the RECTANGLE. The OVAL can be used as the background for other objects or can be used to represent items such as tanks or ponds. By manipulating the object's height and width, the object can be made to have a more circular or more oval shape. The object can be drawn with or without a border. Features such as the background color and border color can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	SOVAL	Object's name
ON (1)	N	Not applicable
OFF (0)	N	Not applicable
INIT	N	Not applicable
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
borderWidth	integer (pixels)	Width of border
borderColor	color name or 6-character HEX	Color of border
backgroundColor	color name or 6-character HEX	Object's background color
fillBackground	Y or N	Y fills object with color specified in backgroundColor; N makes object transparent
drawBorder	Y or N	Y draws border around object; N turns off border

PATTERN (STATIC)

Description	Example
This rectangular object features eight distinct backgrounds (checker, brick, sand, grass, diamond, steel, rain, and earth) and can be used to add visual interest to your screens. This object can be configured as a link to another screen or an HTML page.	 (Brick)
	 (Checker)
	 (Diamond)

	 (Earth)
	 (Grass)
	 (Rain)
	 (Sand)
	 (Steel)

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	PATTERN	Object’s name
patternType	CHECKER, BRICK, SAND, GRASS, DIAMOND, STEEL, RAIN, EARTH	Pattern's design. Type the name of the desired pattern in this field (CHECKER, BRICK, SAND, GRASS, DIAMOND, STEEL, RAIN, or EARTH).
patternSize	integer	Size of the pattern's features; larger numbers increase size of individual rain drops, bricks, sand particles, etc; smaller numbers decrease their size
patternSpacing	integer	Spacing between the pattern's features; larger numbers increase the space between individual rain drops, bricks, sand particles, etc; smaller numbers make the features appear closer together
backgroundColor	color name or 6-character HEX	Object’s background color; for BRICK pattern, this is the color of the mortar between the bricks
URL	N or valid URL address or screen	If a URL address or screen name is entered, the

Field Name	Value	Description
	name	specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
foregroundColor	color name or 6-character HEX	Color of the object's features (for example, the rain drops, particles of sand, blades of grass)
color2	N	Not applicable
angle	integer between 0 and 360	Angle of the object's features; used to make rain drops display at a slant or make bricks vertical instead of horizontal
animate	Y or N	Used with RAIN and SAND patterns only; when set to Y, rain appears as if it is falling and sand appears as if it is being blown in the wind

RECTANGLE (STATIC)

Description	Example
Rectangular object that can be used as a building block for screens. It can be used as the background for an entire screen or to create structures such as buildings or to represent grass or the sky. It can be configured with or without a border. Features such as background color, border color and width can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	SRECT	Object’s name
ON (1)	N	Not applicable
OFF (0)	N	Not applicable
INIT	N	Not applicable
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object

Field Name	Value	Description
heightDim	integer (pixels)	Height (vertical size) of object
borderWidth	integer (pixels)	Width of border
borderColor	color name or 6-character HEX	Color of border
backgroundColor	color name or 6-character HEX	Object's background color
fillBackground	Y or N	Y fills object with color specified in backgroundColor; N makes object transparent
drawBorder	Y or N	Y draws border around object; N turns off border

ROUND RECTANGLE (STATIC)

Description	Example
Rectangular object that features rounded corners. The amount of curve at the corners can be adjusted to make the object appear more square or more round. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	SRRECT	Object's name
ON (1)	N	Not applicable
OFF (0)	N	Not applicable
INIT	N	Not applicable
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
borderWidth	integer (pixels)	Width of border
borderColor	color name or 6-character HEX	Color of border
backgroundColor	color name or 6-character HEX	Object's background color
fillBackground	Y or N	Y fills object with color specified in

Field Name	Value	Description
		backgroundColor; N makes object transparent
drawBorder	Y or N	Y draws border around object; N turns off border
radius	integer	Larger numbers make the rectangle's corners rounder; Smaller numbers make the rectangle's corners more angular

TEXT (STATIC)

Description	Example
Simple text object that can be used as a label or a short block of text. Up to 4 lines of text with up to 12 characters on each line can be accommodated. Features such as text size and color, and background color can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	STEXT	Object's name
line 1	text string	First label to display; up to 12 characters
line 2	text string	Second label to display; up to 12 characters
line 3	text string	Third label to display; up to 12 characters
line 4	text string	Fourth label to display; up to 12 characters
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object
insetX	integer (pixels)	Left or right margin (depending on the horizontal setting specified in justify) as measured from edge of object to label
insetY	integer (pixels)	Top or bottom margin (depending on the vertical setting specified in justify) as measured from edge of object to label
justify	2-character text string	Vertical and horizontal alignment of the label with respect to the object; horizontal alignment is given first. Horizontal values: C=center; R=right; L=left.

Field Name	Value	Description
		Vertical values: M=middle; T=top; B=bottom. (A value of CT would place the text in the center [horizontally] and top [vertically] of the object.)
textColor	color name or 6-character HEX	Color of text
backgroundColor	color name or 6-character HEX	Object's background color
drawBorder	Y or N	Y draws border around object; N turns off border
fontSize	integer	Font size to use for text
fillBackground	Y or N	Y fills object with color specified in backgroundColor; N makes object transparent
bold	Y or N	Text type: Y=boldface text; N=roman (plain) text
concatenate	Y or N	Y links the text strings in lines 1-4 to form one line of text; N places text in lines 1-4 on separate lines

TICK MARK (STATIC)

Description	Example
Similar to a ruler in appearance, this object can be used as a visual measurement of an analog value (such as well level or PSI) when paired with an Analog TREND or Analog SLIDER. Features including the direction and number of tick marks can be adjusted. This object can be configured as a link to another screen or an HTML page.	

Field Name	Value	Description
Value	N/A	Not applicable
ADDR	STATIC	Indicates static-type object
XLOC	integer (pixels)	Horizontal position of object
YLOC	integer (pixels)	Vertical position of object
TYPE	ST	Indicates static-type object
OBJECT	TICKMARK	Object's name
ON (1)	N	Not applicable
OFF (0)	N	Not applicable
image	N	Not applicable
ALARM	N	Not applicable
URL	N or valid URL address or screen name	If a URL address or screen name is entered, the specified custom screen or Web page is opened when the object is clicked (http:// is assumed)
widthDim	integer (pixels)	Width (horizontal size) of object
heightDim	integer (pixels)	Height (vertical size) of object

Field Name	Value	Description
tickColor	color name or 6-character HEX	Color of tick marks
stagger	Y or N	Y draws tick marks in staggered lengths; N draws tick marks in equal lengths
segments	4, 8, 10, or 16	Number of segments (tick marks) on object
direction	L, R, T, B	Direction of tick marks. L=tick marks point right; R=tick marks point left; T=tick marks point down; B=tick marks point up.

Notes

Chapter 6: ALARMS

OVERVIEW

Alarms are an important part of any monitoring and control system. In PMT, an alarm can be created for any physical, logical, or special function register. Several tools are included in PMT for the purpose of monitoring and acknowledging alarms.

The alarm status icon, which appears in PMT's status bar, flashes red when there are unacknowledged active alarms.

Alarm Viewer provides a list of all active alarms (acknowledged and unacknowledged) as well as any cleared alarms that have not been acknowledged.

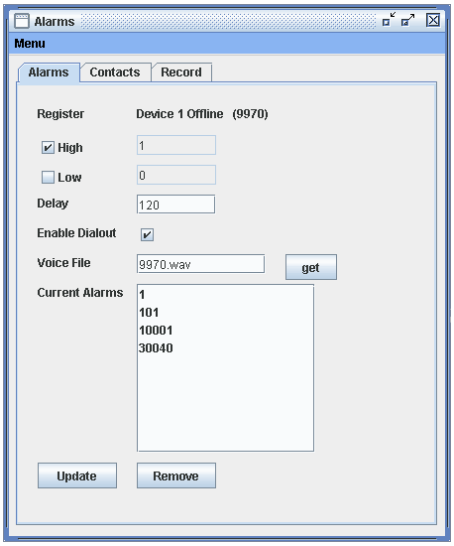
Objects in custom screens can be assigned to a register with a configured alarm. Many objects can be set up to change color or flash when an alarm occurs.

The RDP features an autodialer function. With the autodialer, an alarm can trigger a call to a list of contacts. This enables personnel to be alerted of an alarm condition at an unmanned site.

CREATING ALARMS

Alarms are created using PMT's Alarm Builder. Alarms can be configured for PLC033 and RDP physical, logical, and special function registers. Note however that only the RDP has the dial out function.

1. To open Alarm Builder, select **Alarms** from the **Build** menu.
2. Use Point Picker to select the register the alarm is being created for. The register's tag name and number will be displayed on the Alarms tab next to the word Register. (See "Point Picker Basics" on page 21 for more information on using Point Picker.)
3. Complete the following:



- **High** – Click the checkbox next to **High** if you want the register to alarm when it reaches its "high" state. For digital registers, you only need to select "High"; for analog registers you must also enter the engineering value that represents the high state alarm.
- **Low** –Click the checkbox next to **Low** if you want the register to alarm when it reaches its "low" state. For digital registers, you only need to select "Low"; for analog registers you must also enter the engineering value that represents the low state alarm. If you need an analog register to alarm when its level reaches 0 (zero), enter a small number, for example, .01. This is necessary because entering 0 disables the low alarm.

  NOTE: For digital registers, you can only select one alarm state; for analog registers, you can enter values for one or the other or both states.

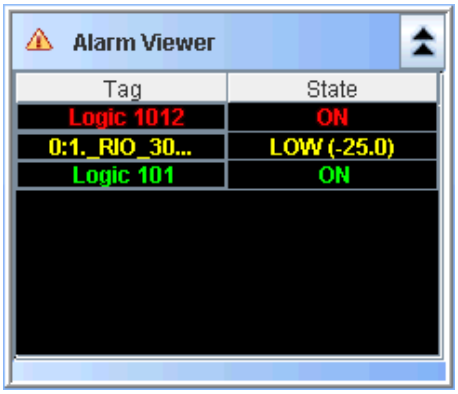
- **Delay** – Length of time (in seconds) the register can be in the alarm state before the system flags it as an alarm. When delay has expired, the alarm will be listed in alarm viewer and dial out (if it has been enabled for this register) will be initiated. Delay can be up to 9,999 seconds (~167 minutes).
- **Enable Dialout** (RDP only) – Select this option if you want the system to call out when this alarm occurs.
- **Voice File** (RDP Only) – Recording that will be played during call out when this alarm occurs. You can create a new recording at this point or use the **Get** button to select a previously recorded file. To "get" a file, click the Record tab and select a file from the **Current Recordings** list. Return to the Alarms tab and click **Get**. The file name will be entered in the **Voice File** box.

4. Click **Update** to add the alarm to the **Current Alarms** list.

   IMPORTANT: Alarms are not saved to the project file until you save the entire project (choose **Save** from the **File** menu).

ALARM VIEWER

Alarm Viewer is a utility that displays a list of alarms that are color coded to indicate their status.



- Red – Alarm is active; alarm has *not* been acknowledged.
- Yellow – Alarm is active; alarm has been acknowledged.
- Green – Alarm has cleared; alarm has *not* been acknowledged.

When PMT is opened, Alarm Viewer is in the tools panel (left panel of the PMT window) between Simulator and Projects.

OPEN, CLOSE AND MOVE ALARM VIEWER

To open Alarm Viewer, click the two downward facing arrows on the top right of the Alarm Viewer window; to close Alarm Viewer, click the two upward facing arrows.

You can also move Alarm Viewer – and the other tools – up and down in the tools panel to rearrange their order.

1. Click Alarm Viewer's title bar. The cursor changes from an arrow to a box.
2. Continue holding down the mouse button and drag Alarm Viewer up or down.
3. Release the mouse button when you have the tool in the desired location.

ACKNOWLEDGE AN ALARM

To acknowledge an alarm in Alarm Viewer, double click the alarm's tag name.

- If it is an active alarm (red), the color will change to yellow.
- If it is a cleared alarm (green), it will be removed from the list.

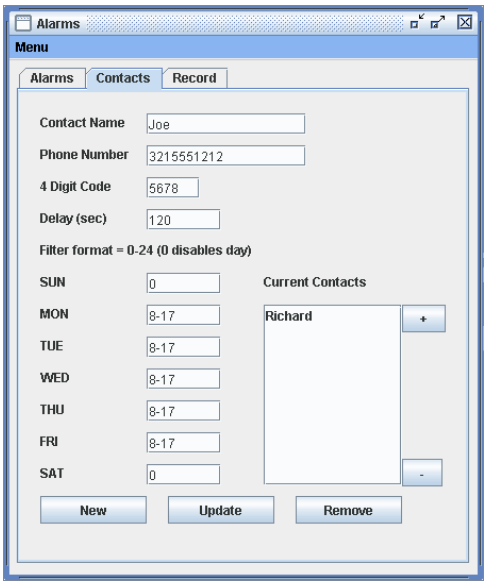
CREATING CONTACTS (RDP ONLY)

Determine the contacts the autodialer should call when an alarm occurs as well as the order they should be called in. The autodialer will call the contacts in the order that they appear in the Current Contacts list. If the alarms aren't acknowledged by the first recipient, the autodialer will move on to the second contact, then the third, etc. If it gets to the end of the list and there are still unacknowledged alarms, it will cycle through the contacts again.

The autodialer can be configured with an unlimited number of contacts to be called when an alarm occurs. Each contact is called in sequence until all of the alarms are acknowledged.

ADD A CONTACT

1. Click **New**.
2. Enter the desired information in all of the Contact form's fields.



- **Contact Name** – Unique name for this contact. The contact name field can hold up to 20 characters.
- **Phone Number** - The phone number field can hold up to 25 characters (hyphens are acceptable, although they are not necessary).

  For functions such as entering an extension number and adding a pause, use the following special characters can be us:

  - Comma (,) – Use to add pauses between the telephone number and an extension number, or between a telephone number and a pager message. Each comma represents a four-second pause.
  - Pound (#) – Use to let a telephone paging system know that the message is complete.
  - Asterisk (*) – Use to gain an outside line or enter an extension number.
- **4 Digit Code** - When an alarm occurs and the autodialer calls out, the recipient of the phone call will be prompted for a four-digit acknowledgement code. The acknowledgement code prevents

unauthorized users from acknowledging alarms. The call recipient will be given three opportunities to enter the correct code before the autodialer will hang up and proceed to the next number in the list. The call recipient will not be allowed to acknowledge any alarms until they enter the correct code.

- **Delay** - The autodialer can be configured with an interval that represents the maximum length of time between phone calls. If the call recipient is unable to acknowledge the alarms within the delay time (for example, if their phone battery dies during the call or no one answers), the autodialer will hang up and dial the next number in the list when the delay time expires. This allows you to set a maximum time that each phone call should take and ensures that alarms will be acknowledged as quickly as possible.
- **Schedule** - By default, each contact is assigned a 24/7 call out schedule. This can be altered by adding filters that tell the system what days and times the contact should be called.
  - For any day that the contact should not be called, enter a 0.
  - To set a range of hours for a day, enter a time span using military format. For example, 7-14 to call a contact between 7:00am and 2:00pm on that day.

3. Click **Update** to add the contact to the **Current Contacts** list.

   IMPORTANT: Contacts are not saved to the project file until you save the entire project (choose **Save** from the **File** menu).

Note that Contacts are listed in the order in which they are added. To move a contact up or down in the list, use the plus (+) and minus (-) buttons to the right of the Current Contacts list.

RECORDING ALARM MESSAGES (RDP ONLY)

Voice recordings are saved locally in the PMT project as .wav files. When you upload (Send) .wav files to the RDP, they are converted to a format (.rmd) that is playable by the RDP's voice modem. Both formats are stored on the RDP. When you download (Get) files from the RDP, only the .wav format file is downloaded.

When voice files are created and saved in PMT or downloaded to PMT from an RDP, the .wav files are stored in the project folder on your hard drive in a subfolder named "voice."

NOTE: The original .wav files must be created in PMT in order for them to be properly converted and function on the RDP. Files created using other software will *not* work.

IMPORTANT: When you save a recording, you are prompted to save it to the server (RDP180). During a single recording upload you may experience a sluggish response in the PMT's user interface for approximately 30 seconds (screens may not update as quickly, alarm status may be delayed). If you will be creating and uploading multiple recordings, it would be better to do it on the bench before the RDP is installed. It is possible to upload all of the recordings at one time using the **Get All Recordings** command. See the next section, "Uploading and Downloading Recordings," for more information.

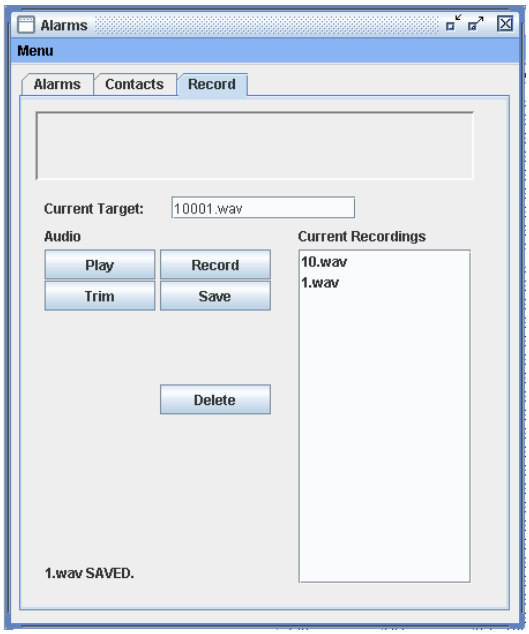
DEFAULT RECORDINGS

The following message are prepackaged with your RDP. You may re-record this messages, but they must be present in order for the autodialer to function properly.

Filename	Message
start	welcome message
entercode	“Please enter your code.”
badcode	“That code is incorrect.”
ack	“Press 1 to acknowledge, 2 to replay.”
alarm	“Alarm.”
allalarms	“All alarms have been acknowledged.”
stillmore	“Not all alarms have been acknowledged.”
goodbye	“Thank you for using the RDP autodialer. Goodbye.”
cleared	“Alarm has cleared.”
ishigh	“Alarm is high.”
islow	“Alarm is low.”
station	“Station.”

RECORD A MESSAGE

1. Open Alarm Builder by choosing **Alarms** from the **Build** menu.
2. Click the **Record** tab.



3. Enter a name for this recording in the **Current Target** box, or use the default file name x.wav, where x is the register associated with the alarm.
4. Click **Record** and begin speaking into the microphone. Notice that the **Record** button turns red and displays the text **Stop**. The message *Recording xxx.wav** appears at the bottom of the Alarm Builder window.
5. When you have completed speaking your message, click **Stop**. The message "Not Saved" is displayed next to the **Current Target** box. The message *xxx.wav recorded to memory** appears at the bottom of the Alarm Builder window.
6. Click **Play** to listen to the recording. If you are not satisfied with the recording, repeat steps 4-6.
7. To trim the recording, use your mouse cursor to highlight the portion of the wave form you want to remove and click **Trim**. This is useful if there is silence at the beginning or end of the recording. Silence appears as a straight line in the wave form box.
8. When you are satisfied with the recording, click **Save**. The file name of the recording is added to the **Current Recordings** list. The message *xxx.wav saved** appears at the bottom of the Alarm Builder window and you are prompted to upload the recording to the server.**
9. Repeat steps 3-8 to record additional messages.

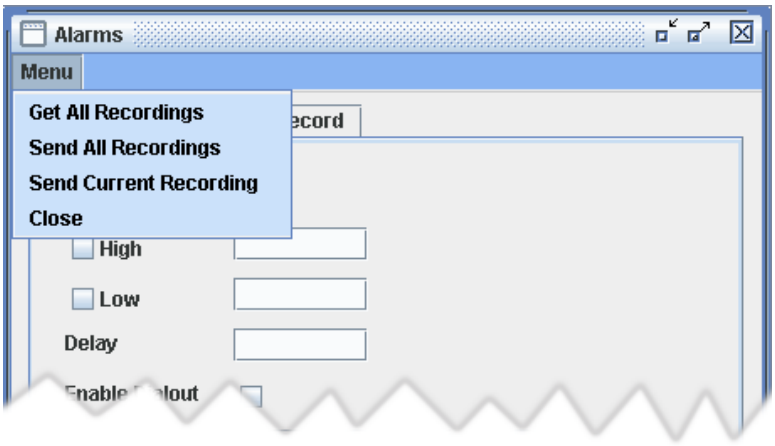
* xxx.wav represents the file name entered in the **Current Target** box.

** During a single recording upload you may experience a sluggish response in the PMT's user interface for approximately 30 seconds (screens may not update as quickly, alarm status may be delayed). If you

will be creating and uploading multiple recordings, it would be better to do it on the bench before the RDP is installed.

UPLOADING AND DOWNLOADING RECORDINGS

Alarm Builder's menu includes commands for downloading voice file recordings from the RDP to PMT and uploading all or a single recording from PMT to the RDP.



GET ALL RECORDINGS

This command is used to download all of the voice files stored on the RDP. Files are downloaded from the IP address specified in Settings (File -> Settings). The downloaded .wav files are placed in the project folder in a subfolder named "voices."

It is important to note that this command will overwrite any existing files in the project that have the same file name.

SEND ALL RECORDINGS

This command is used to upload *all* of the project's voice files to the RDP. Files are sent to the IP address specified in Settings (File -> Settings). The uploaded files are converted to .rmd files – a format recognized by the voice modem. Both formats (.wav and .rmd) are stored on the RDP.

It is important to note that this command will overwrite any existing files on the RDP that have the same file name.

IMPORTANT: It is better to complete this process before installing the RDP. During the conversion process, the responsiveness of the PMT user interface will be severely diminished (the status of custom screens will not update as quickly, alarm status will be delayed). If you will be creating and uploading multiple recordings, it would be better to do it on the bench before the RDP is installed in its permanent location.

SEND CURRENT RECORDING

This command is used to upload a single voice file to the RDP. The file is sent to the IP address specified in Settings (File -> Settings). The uploaded file is converted to an .rmd file – a format recognized by the voice modem. Both formats (.wav and .rmd) are stored on the RDP.

It is important to note that this command will overwrite any existing files on the RDP that have the same file name.

IMPORTANT: During a single recording upload you may experience a sluggish response in the PMT's user interface for approximately 30 seconds (screens may not update as quickly, alarm status may be delayed). If you will be creating and uploading multiple recordings, it would be better to do it on the bench before the RDP is installed. It is possible to upload all of the recordings at one time using the **Get All Recordings** command. See the previous section for more information.

Notes

Appendix A: PLC FIRMWARE UPDATING

From time-to-time, we will make updates to the PLC033 and RDP's firmware (the ROM-based software that controls the device). These updates may add new features to the operation of the device or may fix bugs in the previous release. When updates are available, they will be placed on our server and can be downloaded and installed on the device via the **Update** command in the Process Management Toolkit.

To update the device's firmware, you must be on a computer that has access to the device you want to update. An Internet connection is optional but preferable. If you are connected to the Internet, you are ensured of getting the latest firmware updates. Without an Internet connection, PMT will update the device with any applicable updates that have been stored locally from a previous download. Files are stored locally in a folder in the main PMT directory.

When you initiate the update process, PMT attempts to connect to our server. After successfully connecting to our server, it compares the files on our server to those stored in the local PMT directory. Any updates not found in the local folder are downloaded. If no Internet connection is found, only the locally stored files will be used to update the device. After downloading the updates, the PMT connects to the device, compares the device's current firmware to the locally stored update files, and then makes the necessary updates.

Firmware updates are typically 1-2Mb, although the file size could be smaller or larger depending on the changes that have been made.

FIRMWARE UPDATE PROCEDURE

Note: If you will be updating more than one RDP, a connection to the Internet is only required for the first device. Once the update files have been downloaded to the PMT folder, you can update the remaining devices without being connected to the Internet; they can be updated using the locally-stored files.

1. Start PMT and open the project file for the device you want to update.
2. Select **Update PLC/RDP** from the **Network** menu. The dialog box will show the download progress (to the PMT folder).
3. When the updates have finished downloading, a second dialog box will show the file upload progress (to the device). When the transfer is complete, click **Done**.
4. Reboot the device (cycle power) to finalize the updates.

Notes

Appendix B: RNA FIRMWARE UPDATING

From time-to-time, we will make updates to the RAIL Network Adapter's firmware (the ROM-based software that controls the device). These updates may add new features to the operation of the device or may fix bugs in the previous release. When updates are available, they will be placed on our server and can be downloaded and installed on the device via the Update RNA command in the Process Management Toolkit.

You will be contacted by our Service Department when firmware updates are issued.

1. Launch PMT and select Update RNA from the Network menu.
2. Enter the RNA's IP address and click OK. PMT will connect to our FTP server. It will compare the firmware currently installed on the RNA to the latest available on the server. If updates are available, the files will be downloaded. The dialog box will show a series of files being downloaded. When download is complete, the dialog box's Cancel button will change to read Done.
3. Click Done to close the dialog box.
4. Reboot (cycle power) the RNA to complete the update.

Notes

Appendix C: RELEASE NOTES

Releases are listed in reverse chronological order with the most recent listed first.

PMT Version 2.1.0
Released June 8, 2011

Version 2.1.0 addresses the following:

- Improve initial connection behavior.
- Improve touch terminal behavior.
- Add register tag names to the radio map.
- Add Radio Map export tool so radio configurations can be imported into HT3 station configurations.
- Add Special Function Registers for Cellular Modem statistics.
- Reorder AMM registers in the I/O builder to list AI points first and then DI points.
- Increase register name field in Logic Builder to 40 characters.

Features added/supported in version 2.1.0:

- When PMT connects, it loads active screen data first. The interface is available for use while the initial load continues in the background. (No more hourglass icon when connecting large projects.)
- The sensitivity level of some touch screens results in mouse drag activity, negating the mouse click function. PMT now reacts to mouse down rather than mouse click, improving the behavior with touch screens.
- The Radio Map Tool is executed from the file menu in the Mapper, which creates the file radioexport.xml in the project folder. The XML file contains all module and point data for everything configured in the radio map, including emulated radio addresses. HT3 version 3.1.0 adds a command line radio import tool that reads the XML file and writes the configurations to the database. HT3 version 3.1.0 required. (This procedure must be performed by DFS personnel; refer to Radio Import readme for instructions).
- Three new SFRs have been added to contain cell modem connection data
    - 49945 – Cell Signal Strength – -dbm range from 79 to 118 (displayed as positive number)
    - 49946 – Cell Connection Count – number of successful connections to HT3 driver (resets to 0 upon reboot)
    - 49947 – Cell Error Count – number of failed connections to HT3 driver (resets to 0 upon reboot)

Installation notes

Version 2.1.0 installs over the previous version of PMT 2.0. It isn't necessary to uninstall the previous version first.

Version 2.0.8
Released April 29, 2010

Version 2.0.8 addresses the following:

- Add pulse point configuration for DFS modules in the I/O Builder.
- Automatically reconnect to the PLC/RDP when in full screen mode.
- Upgrade to java Runtime Environment version 76

Features added/supported in version 2.0.8:

Two new fields have been added to module configuration in the I/O Builder to configure pulse points:

- P_Count – the number of pulse points on the module.
- P_Start – The starting register for the pulse points.

Note: Pulse points on the PLC033 are accumulated in an analog register, which will increment from 0-4095. If the value exceeds 4095, the register rolls over to 0 and begins accumulating again. The PLC033 does not retain the value of the register through reboots.

Installation notes

Version 2.0.8 installs over the previous version of PMT 2.0. It isn't necessary to uninstall the previous version first.

Version 2.0.6
Released April 28, 2008

Version 2.0.6 addresses the following:

- PMT does not provide an error message if the path to save the project cannot be found. This can occur if the project is loaded and/or worked on from a shard network folder and the network connection is lost.
- When controlling within the Logic Builder (Force / Ctrl-T), the initial value field in the Logic Inspector changes to the forced value.

Features added/supported in version 2.0.6:

In addition to adding an error message if the project path cannot be found, a new option has been added to the project menu to create a local backup of the project. A new subfolder ("backups") for storing local backups has been added to the PMT program folder. The backup project name will contain the original project name with added time and date stamp. For example, C:\Program Files\PMT\backups\My_Project 2009_04_28 11.33.06. Each time a backup is done, a new project copy with time and date stamp will be created in the backups folder.

Installation notes

Version 2.0.6 installs over the previous version of PMT 2.0. It isn't necessary to uninstall the previous version first.

Version 2.0.5
Released April 21, 2008

Version 2.0.5 addresses the following:

- Ladders that had rungs of logic too large to fit on a single page would not print. The printing method was overhauled to correct this and print ladders more efficiently.
- Analog registers were not being properly removed and/or recreated in the points list. PMT now performs a complete rebuild of the points list each time a ladder check or install is performed. This process is reflected with a "busy" progress bar at the bottom left of the PMT status bar.
- Q Points were not being properly added to the logic file. (This fix only applies to the PLC033.)
- A scaling error for analog points on the screen builder was corrected.
- A math error was corrected with the flow totalizer object.
- A condition that caused the PLC/RDP to intermittently crash when used as a Modbus TCP slave device with HT3 was corrected.

Features added/supported in version 2.0.5

- The Radio Map was enhanced to support configuring multiple stations for radio emulation.
- A new field (Slave Device #) was added to the settings section to support the new Modbus serial slave function.
- Logic Builder has a new right click function that displays system variable addresses.
- RNA110's are now updated via PMT. This is accomplished by selecting **Update RNA** from PMT's **Network** menu. This works similarly to the RDP/PLC update function. PMT obtains the update from the DFS/OCS FTP server and then prompts the user for the IP address of the target RNA110 before installing the update.

Installation notes

Version 2.0.5 installs over the previous version of PMT 2.0. It isn't necessary to uninstall the previous version first.

Version 2.0 (Initial Release)
Released December 20, 2006

PMT 2.0 was developed to enhance the PMT user interface and create more efficient communications between PMT and attached RDP/PLC products. PMT 2.0 replaces PMT version 1.0.

Important: PMT 2.0 will not work with the PLC033 or the RDP033 until they have been updated to System Release 122006.

Features added/supported in this version:

1. Point picker now categorizes registers and provides a grid view to show used and available registers.
2. New simulator panel allows viewing and setting of the values of configured registers. When not connected to the RDP/PLC, the simulator can be used to test logic.
3. Alarm Builder and Viewer have been added to provide alarm handling with PMT. This includes the new callout feature available in the RDP180.
4. Recent projects can be accessed and opened from the new projects panel.
5. New full screen mode has been added with password protection and touch screen support.
6. Software updates for the PLC/RDP can be retrieved from the DFS/OCS FTP server and installed on the device.

Installation notes

PMT 2.0 installs to a different path and does not remove the previous version. If images other than what is distributed in PMT 2.0 are used in existing projects, they will need to be copied from C:\Program Files\console\images to C:\Program Files\PMT\images.

Once PMT 2.0 has been installed and tested, shortcuts to the previous version of PMT should be deleted from the desktop and from the Start Menu.

Appendix D: SUPPORT, SERVICE, AND WARRANTY

SUPPORT AND SERVICE

Please review the information in the troubleshooting section of your product's installation and operation manual before contacting us. If you need further assistance, refer to your product's installation and operation manual for information on obtaining technical assistance via e-mail or phone.

- ***PLC033 Installation and Operation Manual*** (DFS-00507-011-01)
- ***RDP Installation and Operation Manual*** (DFS-00510-011-01)

RETURN AUTHORIZATION (RA) PROCEDURE

The PLC033 and RDP are designed to be robust and highly reliable. We back this performance with a warranty (see our warranty statement for details). In the event that a device fails, during or after the warranty period, it may be returned to be repaired or replaced.

All RA's will be subject to standard shipping and handling charges. Minimum handling charge will be assessed, in most cases, for work such as Radio Tuning, Backplanes, "No Problem Found," and other minor repairs. Handling charges will be waved on warranty equipment. Standard shipping and charges will be based on UPS ground, please advise if other arrangements are needed (UPS Red, FedEx, Pickup, Freight…). Standard cost of repairs and shipping charges can be obtained by contacting our RA Department by phone or e-mail.

Refer to the installation and operation manual for your device for information on returning a device.

- ***PLC033 Installation and Operation Manual*** (DFS-00507-011-01)
- ***RDP Installation and Operation Manual*** (DFS-00510-011-01)

WARRANTY

Refer to the installation and operation manual for your device for information on its warranty.

- ***PLC033 Installation and Operation Manual*** (DFS-00507-011-01)
- ***RDP Installation and Operation Manual*** (DFS-00510-011-01)

QUESTIONS OR COMMENTS ON THIS MANUAL

If you find a problem with any of the information in this manual or have suggestions on how it could be improved, please contact us at the address below:

Data Flow Systems, Inc.
Documentation Department
605 N. John Rodes Blvd.
Melbourne, FL 32934

Alternatively, e-mail us at:

documentation@dataflowsys.com

Data Flow Systems, Inc.
605 N. John Rodes Blvd.
Melbourne, FL 32934
321-259-5009
www.dataflowsys.com